



**HAL**  
open science

## Visualization of Biomolecular Structures: State of the Art Revisited

B. Kozlíková, M. Krone, M. Falk, N. Lindow, Marc Baaden, D. Baum, I. Viola, J. Parulek, H.-C. Hege

► **To cite this version:**

B. Kozlíková, M. Krone, M. Falk, N. Lindow, Marc Baaden, et al.. Visualization of Biomolecular Structures: State of the Art Revisited. *Computer Graphics Forum*, 2017, 36 (8), pp.178-204. 10.1111/cgf.13072 . hal-01400465

**HAL Id: hal-01400465**

**<https://hal.science/hal-01400465v1>**

Submitted on 13 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Visualization of Biomolecular Structures: State of the Art Revisited

B. Kozlíková<sup>†1</sup>, M. Krone<sup>†2</sup>, M. Falk<sup>3</sup>, N. Lindow<sup>4</sup>, M. Baaden<sup>5</sup>, D. Baum<sup>4</sup>, I. Viola<sup>6,7</sup>, J. Parulek<sup>6</sup>, and H.-C. Hege<sup>4</sup>

<sup>1</sup>Faculty of Informatics, Masaryk University, Czech Republic

<sup>2</sup>Visualization Research Center, University of Stuttgart, Germany

<sup>3</sup>Immersive Visualization Group, Linköping University, Sweden

<sup>4</sup>Department of Visual Data Analysis, Zuse Institute Berlin, Germany

<sup>5</sup>Laboratoire de Biochimie Théorique, UPR 9080 CNRS, France

<sup>6</sup>Department of Informatics, University of Bergen, Norway

<sup>7</sup>Institute of Computer Graphics and Algorithms, TU Wien, Austria

---

## Abstract

*Structural properties of molecules are of primary concern in many fields. This report provides a comprehensive overview on techniques that have been developed in the fields of molecular graphics and visualization with a focus on applications in structural biology. The field heavily relies on computerized geometric and visual representations of three-dimensional, complex, large, and time-varying molecular structures. The report presents a taxonomy that demonstrates which areas of molecular visualization have already been extensively investigated and where the field is currently heading. It discusses visualizations for molecular structures, strategies for efficient display regarding image quality and frame rate, covers different aspects of level of detail, and reviews visualizations illustrating the dynamic aspects of molecular simulation data. The survey concludes with an outlook on promising and important research topics to foster further success in the development of tools that help to reveal molecular secrets.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

---

## 1. Introduction

Interactive molecular visualization is one of the oldest branches of data visualization [Lev66, Fra02], with deep roots in the pre-computer era. This paper reviews interactive visualization of *biomolecular* structures – the subfield that developed most during the past two decades. This paper is an extended version of our previous survey of molecular visualization [KKL\*15] and includes newer work that was not available at that time. This revised version also includes references that are historically interesting and provided the foundations for the current state of the art.

First, let us characterize the objects of interest. Ordinary matter consist of atoms and molecules, which in turn embody protons, neutrons, and electrons. The protons and neutrons are bound together by nuclear forces, forming the nuclei of the atoms. The positively charged nuclei attract negatively charged electrons; due to quantum mechanical effects the particles do not collide, but the electrons surround

the nuclei in defined distances, comprising stable and electrically neutral *atoms*. These are the smallest units of a chemical element. The electrons in an atom are organized in orbitals, i.e., regions of space, in which electrons stay with high probability. Each atomic orbital can contain up to two electrons, whose energy, angular momentum, and magnetic moment determine the orbital's shape and location. In a simplified model, atomic orbitals are arranged in layered shells and sub-shells. The outer electrons of two atoms can interact and form molecular orbitals, potentially creating a *chemical bond* between the atoms. Bonds are classified as being either strong (covalent, ionic, and metallic bonds) or weak (dipole-dipole interactions and hydrogen bonds). Due to strong bonds, sets of atoms are held together, forming tight entities like molecules, ionic salts, and metals. A *molecule* thus is a structure composed of nuclei, defining the atom positions, and core electrons (inner electron shells); the nuclei are held together by an outer electronic shell (valence shell), composed of molecular orbitals. Molecules are the smallest units of a compound, i.e., of a pure chemical substance. Molecules playing an active role in living systems are called *biomolecules*. These include large molecules

---

<sup>†</sup> These authors contributed equally

(macromolecules) such as proteins, polysaccharides, lipids, DNA and RNA, as well as small molecules such as metabolites. Weak bonds occur inside molecules as well as between molecules. They are critical in maintaining the 3D structures of biomolecules, in forming larger entities (molecular complexes) and in binding molecules specifically but transiently, creating thereby the basis of many biological processes.

The primary purpose of *molecular visualization* is to support our understanding of the rich, complex material world, by making molecular structures, their properties, and their interactions apprehensible and intelligible. In addition it aims at supporting the ‘rational’ design of new molecules, such as pharmaceutically active compounds, or customized substances with specific properties. The subfield *biomolecular visualization* deals with the graphical depiction of the structure, interaction and function of biomolecules, biomolecular complexes, molecular machines, and entire biological functional units that occur in biological cells. Additionally, it complements the toolset of bioinformatics by providing means for integrated visual analysis of sequence and structure data, e.g., for elucidating phylogenetic trees; examples are viewers displaying families of closely related structures, mutants, and conserved structural motifs.

Forerunners of today’s visual representations of atoms and molecules are *hand-drawn depictions* and *physical models*. Pictorial representations have been used, e.g., by Kepler (1611) [Kep11] and Huygens (1690) [Huy90], centuries before 1808, when Dalton published the modern, but still pre-quantum formulation of atomic theory [Dal10]. In these groundbreaking works, atomic arrangements were illustrated, displaying atoms as spheres. Van der Waals (1873) [vdW73], saw the necessity of taking into account the molecular volume as well as attracting intermolecular forces; he computed from experimental data the volume occupied by an individual atom or molecule. From now on, approximate atomic radii for several chemical elements were known and used in depictions. Also physical models of molecules, both static and dynamic, have been used for visualization purposes. They allowed to correlate mechanical, optical, electrical, magnetic, and thermal properties with the atomic structure and to reproduce atomic and molecular movements. For a survey covering the period 1880–1960 see [Smi60].

With the emergence of increasingly elaborate atomic models by Thompson, Rutherford, Bohr, and Sommerfeld in the early 19th century, more detailed visualizations became necessary, culminating in detailed depictions of complex atoms in the popular book of Kramers and Holst [KH23], showing the elliptic orbits of electrons in the Bohr-Sommerfeld model. However, in these years it became clear that atoms and molecules are of truly quantum nature. Quantum physics, however, seems to be intrinsically *non-visualizable*. One of several reasons is that no (mental) image exists that simultaneously represents the corpuscular

and wave-like character of particles. According to Heisenberg’s uncertainty relation, an electron cannot be considered to have an exact location in its orbital, i.e., its trajectory is not defined [Hei26]. Instead, according to Born [Bor26], an electron’s position is described by a probability distribution, given by the absolute square of Schrödinger’s complex wave function  $\Psi$ . The evolution of  $\Psi$  for a system of  $N$  quantum particles, described by the time-dependent Schrödinger equation [Sch26], happens not in real 3-dimensional space, but in  $3N$ -dimensional space of all particles’ coordinates. This poses a further challenge to visualization. Regarding visualization of fully quantum physical systems only very limited work is available; examples are [Tha05, BD12].

Fortunately, research revealed that molecular systems can be *classically* described to a good approximation, if no covalent bonds are newly formed or broken, and if the system’s behavior does not depend sensitively on fine-tuned energy values. In *molecular dynamics (MD) simulations*, no molecular orbitals are computed; instead atoms are treated as classical objects that move under the influence of artificial multi-body forces (‘force fields’) that mimic quantum effects. Due to the strong repulsion between neutral atoms and molecules, atoms can be considered approximately as ‘hard’ spheres. This means, atoms are fully characterized by their mass, radius, and the multi-body forces they exert on other atoms, ‘inner’ electronic degrees of freedom are neglected. The majority of MD simulations, particularly of biomolecular structures, is performed using this ‘classical’ approximation. The depiction of van der Waals spheres thus was one of the starting points of modern molecular computer graphics, beginning with the work of Lee and Richards (1971) [LR71]. This work has been continued, now for more than four decades, with the invention of further types of molecular surfaces representing the spatial accessibility of molecules.

However, some types of biological systems require quantum mechanical considerations for a detailed understanding. Examples of biological and medical relevance are enzymatic reactions or photosynthesis. See, e.g., [ADP08, AKM14] for popular-science presentations of the emerging ‘quantum biology’. This opens up a new field of research in molecular visualization, on which we will report only very briefly.

In the next section, the basics of biomolecular data are outlined, including data sources. Section 3 introduces a taxonomy of the literature about molecular visualization covered by this report and gives an overview of the structure of the rest of the paper (Sections 4 to 6). The report is concluded by a brief overview of molecular visualization tools (Section 7) and anticipated future challenges (Section 8).

## 2. Molecular Data

This section introduces the input data, mostly formed by biomolecules, along with their composition and basic properties. Moreover, the most common sources of molecular structures and molecular dynamics are discussed.

## 2.1. Biomolecules

Biomolecules usually carry out important functionality including enzymatic catalysis, coordinated motion, mechanical support, immune protection, generation and transmission of nerve impulses, and reproduction [Str95]. Some of these molecules are rather large entities and are, therefore, referred to as macromolecules. Others are building blocks of complex structures such as membranes. The majority of small biomolecules takes an active role in the metabolism of an organism and are hence called *metabolites*. Below, the most important types of biomolecules are briefly introduced.

The building blocks of *nucleic acids* are *nucleotides*, which are composed of a nucleobase, a sugar, and a phosphate group. The main difference between deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) is that the sugar in DNA is a deoxyribose, while it is ribose in RNA. Further, one of the four bases occurring in DNA, thymine, is replaced by uracil in RNA. DNA usually forms the characteristic double helix of two single DNA strands first identified by Watson and Crick [WC53]. In contrast, RNA is single-stranded and typically forms very complex structures by folding onto itself. DNA stores the genetic code including the information about the composition of the proteins; both DNA and associated proteins may be epigenetically modified to regulate gene expression. During protein synthesis, the respective part of the DNA is transcribed into RNA, which is in turn translated into the amino acids that form the protein.

*Proteins* are macromolecules consisting of one or more chains of *amino acids*. Different proteins have diverse functions like replication of DNA, catalyzing chemical reactions, or transport of other molecules. The amino acids forming the protein chain are connected via peptide bonds. This chain is called the protein's primary structure. The amino acid chain folds into an energetically favorable configuration stabilized by intramolecular interactions, such as hydrogen bonds. The folding introduces patterns to the protein chain called secondary structure. The two most common secondary structure elements are the  $\alpha$ -helix [PCB51] and the  $\beta$ -sheet [PC51], which are connected by loops and unstructured parts called random coil. The correct folding of the chain is important for the proper function of most proteins. The three-dimensional arrangement of the secondary structure of the protein chain is called tertiary structure. Two or more folded chains can form a functional complex called quaternary structure. In the visualization literature, the term secondary structure sometimes is used synonymously for tertiary and quaternary one, see, e.g., [WB11]. In addition, proteins may undergo a post-translational modification during their maturation.

*Lipids* and *lipid membranes* are ubiquitous in biological systems as they delineate the compartments of the cell, control entry and transport, and harbor important membrane proteins. In addition to lipids, proteins and nucleic acids, cells contain *sugar molecules* carrying out crucial biological functions and storing energy. Sugars may attach to proteins

or lipids and form extremely complex polymers, the polysaccharides. Many *small molecules*, *metabolites*, and *ions* are further central ingredients necessary for life [Goo09]; actually they are frequently present and important in structural data. A few examples include energy-providing ATP, electron-transporting NAD and other prosthetic groups.

## 2.2. Molecular Structure Acquisition

In-vitro experiments provide a key resource for molecular structural data based on the following three techniques: *X-ray crystallography* [Woo97], which potentially leads to the highest resolution data when crystals can be obtained; *nuclear magnetic resonance spectroscopy* (NMR) [Wüt86] determining structural ensembles rather than a single structure; *cryo-electron microscopy* (cryo-EM) [vHGM\*00] allowing the determination of large structures, but requiring an image-based reconstruction with limited resolution.

*Molecular simulation* is a useful method to study the dynamic behavior of previously determined molecular structures. It allows scientists to study the effect of different parameters (like temperature, solvent type, and concentration) and the interaction with other molecules. The most frequently used methods are Monte Carlo (MC) sampling and molecular dynamics (MD) simulations. An overview of these methods can be found in the textbooks by Frenkel and Smit [SF02] and Schlick [Sch10]. Both methods usually do not model quantum mechanical effects explicitly and incorporate such effects only through the classical molecular force fields being used. Hybrid MC methods have been developed to combine the merits of both methods. If the molecular systems to be simulated become very large (several million to billion of atoms), it is computationally very expensive to simulate the system for relevant time intervals of milliseconds or even seconds. Although Shaw et al. [SGB\*14] demonstrated that it is possible to run ribosome-sized simulations of a few million atoms at multiple microseconds per day, in most cases it is still necessary to abstract from atomic resolution and move to coarse-grained models. In these models, groups of atoms instead of single atoms are considered as the smallest unit. Depending on the molecular systems, several types of coarse-grained models can be adopted (see, e.g., [Cle08]). Recently, Krieger and Vriend [KV15] introduced a set of algorithms aiming to improve the performance of MD simulations. If a simulation process is mainly controlled by diffusion, Brownian Dynamics (BD) is often used as a complementary approach to MD [AM06].

The results of molecular modeling and simulation methods are trajectories of coordinates of particles. In the case of all-atom simulations, these particles are atoms while for coarse-grained simulations, each particle represents the center of mass of a molecule or a group of atoms.

In contrast to the molecular simulation techniques mentioned above, normal mode analysis (NMA) calculates large-amplitude molecular motions without simulating the motion

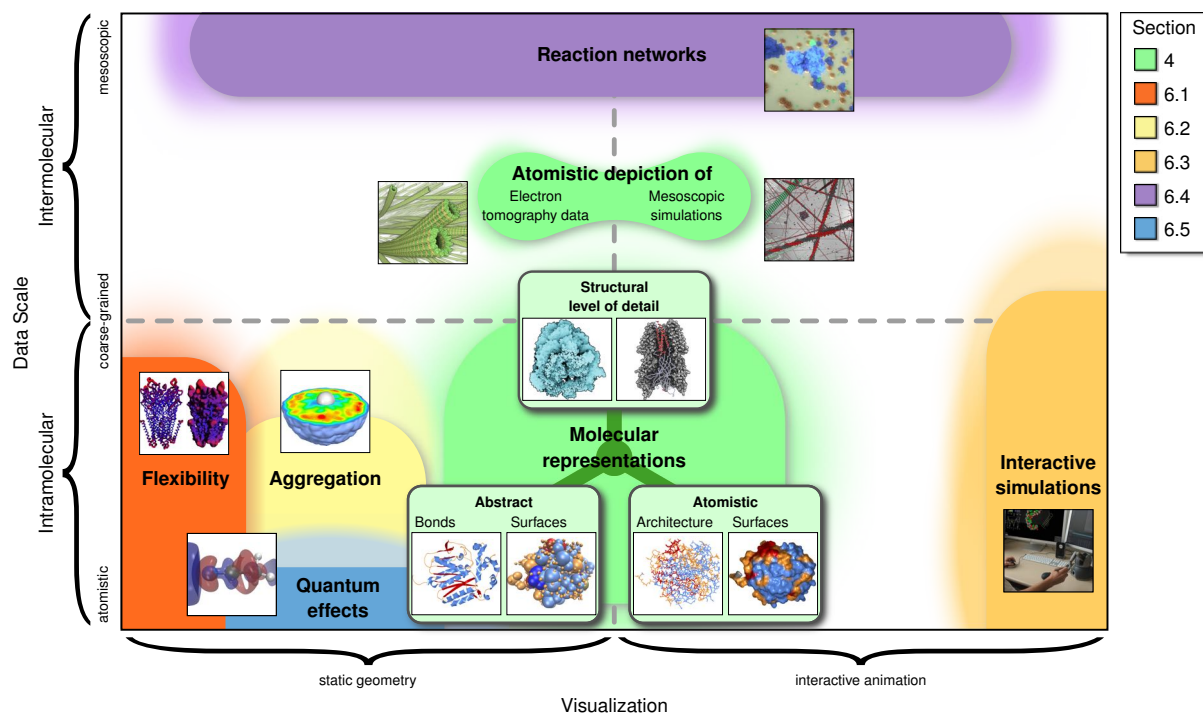


Figure 1: Illustrated taxonomy of the literature about molecular visualization covered by this report. Miniatures created by or taken from [CDB\*15,FKRE10,FKE13,HDS96,KSS\*14,LBH12,LTDS\*13,MWPV15,PJR\*14,vdZLBI11].

of a molecule [BR05]. It is much faster than classical molecular simulation and, thus, allows the study of large-scale macromolecular motions taking place at a long time scale, while trading accuracy.

Recently, Johnson et al. developed a semi-automatic modeling tool called *cellPack* [JAAA\*15] that computes a packing of molecules to form comprehensive models of very complex molecular systems up to mesoscopic length scales.

Another source of data are biochemical reaction models, which can be categorized roughly as kinetic models and particle-based models. Kinetic models are typically described by pathway networks augmented with spatial information at times. In contrast, the focus of particle-based models lies on the action and interaction of individual agents, i.e., the particles. An agent is assigned with a set of rules of how to behave in a certain environment and how to interact with other agents, i.e., other molecules. Popular frameworks for simulating cellular environments with particles include *MCell* [SB01], *ChemCell* [PS05], and *Smoldyn* [AABA10], covering membrane interaction, diffusion, and particle-particle reactions. The computational cost of agent-based simulations is usually very high and time-consuming compared to kinetic models. Another efficient method to study biochemical reaction models is stochastic simulation [Gil07]. Recently it was shown [RSL13] that the chemical master equation and the reaction-diffusion master

equation, both underlying stochastic simulations, can be efficiently sampled on GPUs, speeding up the computation up to two orders of magnitude.

### 3. Taxonomy

Figure 1 depicts the taxonomy that is used to classify the methods covered by this report. We distinguish between four major areas shown as quadrants in the figure. These quadrants are defined by the type of visualization along the horizontal axis and the data scale on the vertical axis. The types of visualization can be subdivided into showing a *static geometry* (left side) or depicting an *interactive animation* (right side). Visualizing static geometry results in a still image. Such an image can nonetheless show dynamic properties or attributes derived from these. The interactive animation on the other hand focuses on the interactive playback to further emphasize features related to dynamics. Instead of showing a pre-rendered movie, the animation is computed and shown on demand. In both cases, the visualization typically allows for interactive adjustment of parameters like camera settings by the user.

The vertical axis corresponds to the scale of the underlying data that is visualized. Although being continuous, this axis can be divided into two major areas with respect to molecular visualization. The *intramolecular scale* ranges

from atomistic data on the atomic scale to coarse-grained molecular models. The *intermolecular scale* covers coarse models up to the mesoscopic level, where entire molecules are considered as a single entity. The actual scale of the data mostly depends on the data acquisition, e.g., molecular structures obtained by NMR or results of mesoscopic intracellular simulations. Please note that coarse data might be enriched in the visualization to add more details. One example of such an augmentation is the replacement of structural data on the intermolecular scale with details on the atomistic scale, i.e., individual atoms [LBH12, FKE13]. Furthermore, additional bioinformatics data like phylogenetic trees and other biomolecular information can be included as well.

The colored areas in Figure 1 correspond to the various concepts discussed in the subsequent sections. Their positions coincide with the type of visualization and data scale where the respective methods and algorithms are typically applicable to. Molecular representation models (green) are described in Section 4. These representations can be divided into atomistic models (Section 4.1), illustrative and abstract models (Section 4.2), and structural level of detail (Section 4.3). They can be applied to visualize static and dynamic attributes on the intramolecular scale. One exception is the depiction of atomistic detail on the intermolecular scale, which utilizes the enrichment described above (cf. Section 4.3). The remaining areas can be summarized under the term of visualization of molecular dynamics (Section 6). This includes the visualization of flexibility (red, Section 6.1), volumetric representations and aggregation (yellow, Section 6.2), interactive and steered simulations (orange, Section 6.3), visualization of molecular reactions (violet, Section 6.4), and visualization of quantum effects (blue, Section 6.5). The techniques for molecular rendering described in Section 5 are not included in the taxonomy, since they are generally applicable to the majority of molecular visualizations.

## 4. Molecular Representation Models

In chemistry, many three-dimensional molecular models have been developed that show different attributes of the depicted molecule. The choice of the molecular model used for data visualization depends on the intended analysis task. The models can be classified into atomistic ones (Section 4.1) and abstract ones (Section 4.2), as is shown in the illustrated taxonomy in Figure 1. Large molecular systems are often depicted using level of detail visualizations (Section 4.3), which include continuous representations as defined by Goodsell [Goo99] that simplify the atomic details.

### 4.1. Atomistic Models

Atomistic models directly depict atoms of a molecule. The atomic structure always plays an essential role in determining molecular properties. The atomistic representations

model discrete entities interacting through pair-wise forces and are usually used in molecular systems consisting of up to millions of atoms. The atomistic models can be further classified into models that focus on the bonds between atoms and surface models, which illustrate the interface between a molecule and its environment.

In traditional interactive molecular graphics, molecular models are typically triangulated, since GPUs are designed for fast triangle rendering. To achieve a reasonable quality, however, often many triangles are required, which can impede interactivity. Since many models can be decomposed into simple implicit surfaces, e.g., spheres and cylinders, modern GPU-based glyph ray casting as presented by Gumhold [Gum03] to render ellipsoids became more efficient. The general idea is to render a projection of a primitive that bounds each implicit surface (i.e., glyph). Then, for each fragment of said primitive, the intersection of the view ray with the implicit surface is computed in the fragment shader. Reina and Ertl [RE05] used a combined ray casting of spheres and cylinders to visualize mono- and dipoles in MD data. Sigg et al. [SWBG06] formulated a general concept for ray casting arbitrary quadrics on the GPU. GPU-based ray casting can still be seen as the current state-of-the-art. It enables rendering a massive number of simple surfaces in real-time with pixel-perfect quality for any zoom level.

#### 4.1.1. Bond-centric models

Visualizing chemical bonds between atoms helps to understand and to predict many chemical properties of the given molecule. Bond-centric models that display the chemical bonds between individual atoms of the molecular system were designed for this purpose. The most often used bond-centric model visualizing only bonds is called *licorice* or *stick* model. The bonds can be augmented with the atoms forming these bonds, which results in a representation called *ball-and-stick*, which is one of the oldest and most often used structural representations.

The simplest representation of bonds is the *lines* model. More sophisticated visualizations represent the bonds by cylinders and atoms by spheres. As described above, modern GPU-based ray casting techniques are much more efficient and achieve higher visual quality than traditional triangle-based rendering for these implicit objects. However, most of the modern techniques for bond representation and van der Waals representation are descendants of techniques and software tools that came out in the late 80s and early 90s [FPE\*89, MEP92].

Chavent et al. [CVT\*11] introduced a novel representation called *HyperBalls*. Instead of the traditional stick representation of bonds, it smoothly connects atom spheres by hyperboloids. Hyperboloids can be defined by a cubic equation, which makes them suitable for GPU-based ray casting.

### 4.1.2. Surface Models

**Space-filling Models and Van der Waals Surfaces.** The simplest and probably most often used molecular model is the *space-filling* or *calotte* model. Here, each atom is represented by a sphere whose radius is proportional to the atomic radius, e.g., covalent radius, of the respective element. The surface is then defined as the outer surface of the union of all atom spheres (blue spheres in Figure 2). The *van der Waals (vdW) surface* [Ric77,GS87] is a space-filling model where the radius of the atom spheres is proportional to the van der Waals radius (Figure 3). This surface shows the molecular volume, that is, it illustrates the spatial volume the molecule occupies. The vdW surface is the basis of most other molecular surface representations. In 1995, Sayle and Milner-White presented the molecular graphics tool *RasMol* [SMW95], which supports the vdW representation and can exploit several CPUs during rendering. They demonstrated that the vdW representation could become a fast visualization technique for molecules. Nowadays, GPU-based ray casting of the vdW spheres is the fastest way to visualize the surface of several million of atoms [GRE09]. Recently, further techniques were proposed to handle even larger data sets (see Section 4.3).

**Solvent Accessible Surface.** Lee and Richards defined one of the first extensions to the vdW surface, which later became known as the *solvent accessible surface (SAS)* [LR71]. The idea of this surface is to show all regions of a molecule that can be accessed by a solvent molecule. To simplify the computation, the solvent molecule is approximated by a single sphere—the probe. The SAS is described by the center of the probe while rolling over the vdW surface, as shown in Figure 2. During this process, the probe always touches the vdW surface but never penetrates it. All points outside the surface can be geometrically accessed by the center of the probe and, thus, probably also by the solvent. Consequently, all atom spheres contributing to the SAS are accessible to a molecule with radius equal to or smaller than the probe radius. This makes the SAS feasible for analyzing possible binding partners or transport channels. The disadvantage of the SAS, however, is that it does not faithfully show the molecular volume since the molecule is inflated. This can lead to intersections with other molecules, e.g., when visualizing a molecular simulation. The SAS is identical with the vdW surface where each vdW radius is extended by the radius of the probe. All visualization techniques for the vdW surface can also be used to render the SAS.

**Solvent Excluded Surface.** In 1977, Richards [Ric77] defined the first smooth molecular surface (see Figure 3) based on the idea of the SAS. Instead of taking the center of the probe that rolls over the atoms, he suggested to use the boundary of the spherical probe (see Figure 2). This combines the advantages of both previous surfaces, the better size representation of the vdW surface and the accessibil-

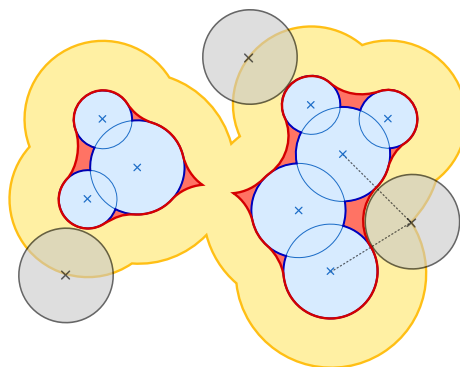


Figure 2: 2D schematic of vdW surface (blue), SAS (yellow), and SES (red). The SAS and SES are defined by a spherical probe (gray) that rolls over the vdW surface.

ity visualization of the SAS. Greer and Bush gave an alternative definition [GB78], which is equivalent to the one of Richards. They defined the surface as the topological boundary of the union of all possible probe spheres that do not penetrate any atom of the molecule. Their work coined the commonly used term *solvent excluded surface (SES)*. Mathematically, the SES is composed of three types of patches: Convex spherical patches occur where the probe touches exactly one atom; toroidal (or saddle) patches are tracks where the probe touches exactly two atoms; concave spherical patches occur where the probe lies in a fixed position, touching exactly three atoms. At the patch boundaries, where two or more patches fit together, the surface is  $C^1$ -continuous, i.e., the SES is smooth. However, the surface can contain self-intersections, which are also called ‘singularities’ [SOS96]. At these intersections the surface has sharp edges and is only  $C^0$ -continuous. Two types of self-intersections can occur when the atoms lie too far away from each other. The first type is the self-intersection of toroidal patches. This type occurs when the probe rolls between two atoms and intersects the axis of revolution through the two atom positions, thereby creating a spindle torus. The second type occurs when two or more concave spherical patches intersect.

The algorithms for computing the SES fall into two categories. The first comprises all methods that compute the surface by discretizing the space  $\mathbb{R}^3$ . These approaches usually compute a discrete scalar field from which an isosurface is extracted, either by triangulation via Marching Cubes [LC87] or by direct isosurface ray marching. Two of the fastest approaches in this research area were presented by Can et al. [CCW06] and Yu [Yu09]. Although these algorithms are typically easy to implement, the computation time and memory requirements increase cubically with the grid resolution. The second category contains all methods that compute an analytical representation of the surface by determining the implicit surface equations of all patches. In 1983, Connolly [Con83] presented the equations to compute the

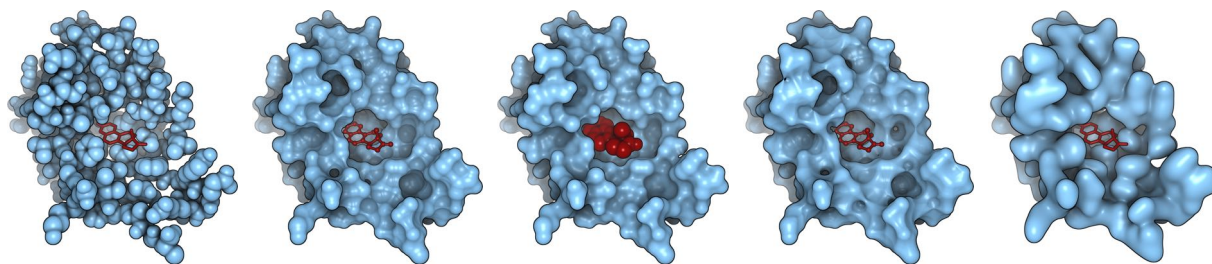


Figure 3: Comparison between different molecular surfaces of the protein isomerase (PDB ID: 1OGZ). From left to right: vdW surface, SES with probe radius 1.4 Å, LES for equilenine, MSS with shrink factor 0.35, and Gaussian surface with standard deviation equal to the atom radius. The ligand equilenine (red) is shown as stick, ball-and-stick, or vdW surface, respectively. In all examples, depth cueing in combination with screen-space ambient occlusion is applied and silhouettes are shown.

SES analytically and the first algorithm based on this. Varshney et al. [VBW94] proposed a parallel algorithm based on the computation of an approximate Voronoi diagram. Edelsbrunner and Mücke [EM94] introduced *alpha shapes* that can be used to compute the SES. Sanner et al. [SOS96] presented the *reduced surface* (RS) algorithm. This algorithm is very efficient but iterative and, thus, not easily parallelizable. The RS can be updated partially in order to support dynamic data [SO97]. In 2009, Krone et al. [KBE09] achieved interactive frame rates for dynamic molecules with a few thousands of atoms using an optimized implementation of the RS algorithm. In 1996, the same year Sanner et al. presented their reduced surface algorithm, Totrov and Abagyan [TA96] proposed the *contour-buildup* (CB) algorithm. It directly computes the track of the probe on each atom surface and therefore is embarrassingly parallel. Lindow et al. [LBPH10] presented a parallel CB algorithm using OpenMP, which allowed the user to visualize dynamic molecules with up to  $10^4$  atoms on 6 core systems. Krone et al. [KGE11] parallelized the CB algorithm for GPUs, which further accelerated the SES computation and enabled the interactive visualization of dynamic molecules with up to  $10^5$  atoms. These two methods are currently the fastest analytical techniques to compute the SES.

For visualization purposes, the SES was traditionally tessellated. Examples for two very accurate tessellations are the one by Sanner et al. [SOS96] and the one by Laug and Borouchaki [LB02]. Later, Zhao et al. [ZXB07] proposed a triangulation that approximates the patches by spline surfaces to simplify the triangulation. One of the fastest methods was proposed by Ryu et al. [RCK09] using subdivision surfaces. Their approach, however, is not able to handle all possible singularities.

Triangulating the SES is computationally expensive and usually takes seconds for mid-sized proteins. In 2009, Krone et al. [KBE09] thus used GPU-based ray casting to render the three types of surface patches. As mentioned above, it yields not only pixel-perfect image quality but is also much faster, even though quartic equations have to be solved.

Krone et al. also handled the self-intersections of the SES patches using ray casting. Lindow et al. [LBPH10] presented a slightly improved ray casting that uses the geometry shader to optimize the rasterization of primitives. This accelerated the rendering by approximately 30%. To optimize the ray casting performance, the parts of the convex spherical patches lying inside the SES were not clipped in the previous methods. Hence, the surface could be visualized only opaquely or with a simple blending of the front face. Semi-transparent or clipped visualizations, however, require a complete clipping of these patches. A solution for this was described by Kauker et al. [KKP\*13]. Ray casting is currently the fastest techniques to visualize the SES while also offering the highest image quality.

In 2012, Parulek and Viola presented the first ray casting of the SES that does not need a pre-computation of the analytical description of the surface [PV12]. They use a modified sphere tracing and directly compute the implicit description of the surface based on the local neighborhood of the ray. This enables the direct visualization of the SES for dynamic molecular data. However, due to the complexity of this extended ray casting, interactive frame rates are only achieved for molecules up to 2,000 atoms. The technique also offers a level of detail strategy that improves the rendering performance, but can lead to pixel artifacts, e.g., at singularities and patch boundaries. Details can be found in the STAR by Patane and Spagnuolo [PS15] on geometric and implicit modeling for molecular surfaces.

Decherchi and Rocchia [DR13] presented a combination of triangulation and ray casting. The algorithm computes the analytical description of the SES and performs a ray casting along a 3D grid from which the surface is triangulated using Marching Cubes. Although they could accelerate the triangulation of the SES, the overall speed and visual quality cannot compete with direct ray casting.

**Molecular Skin Surface.** Edelsbrunner presented a new smooth surface for a finite set of input spheres, called *skin surface* [Ede99]. Its shape depends on a single param-



ter  $s \in (0, 1]$ , the shrink factor. The *molecular skin surface* (MSS) is the application of the skin surface to the vdW spheres of the atoms. The main advantage of the MSS over the SES is that the surface is completely  $C^1$ -continuous (see Figure 3). Furthermore, it can be decomposed into patches of quadrics. However, the MSS has no biophysical background. Kruithof and Vegter [KV07] presented a topology certified tessellation approach for the MSS. Cheng and Shi [CS09] developed a triangulation algorithm that achieves a higher quality but has the disadvantage that it is very time consuming. A very fast triangulation was presented by Decherchi and Rocchia [DR13] following the same strategy as their SES approach. However, it does not necessarily preserve the full surface topology. To achieve a fast, high-quality visualization, Chavent et al. [CLM08] presented the first GPU-based ray casting to render the MSS. The long run times of their implementation for the construction of the MSS, however, prevented the use for dynamic molecular data. In 2010, Lindow et al. [LBPH10] presented an accelerated computation using the same idea that Varshney et al. [VBW94] applied to compute the SES. They also optimized the ray casting of the MSS. As result of both improvements, interactive MSS visualization of dynamic molecules with a few thousand atoms became possible.

**Ligand Excluded Surfaces.** The *ligand excluded surface* is a generalization of the SES (see Figure 3). It was recently proposed by Lindow et al. [LBH14]. In contrast to the SES, the LES does not approximate the ligand by a sphere but uses the full and potentially dynamic geometry defined by the ligand’s vdW surfaces. Thus, the LES shows the geometrical surface that a specific ligand can access when approaching the molecule. An analytical computation of the LES is difficult. Lindow et al. therefore proposed an algorithm to compute the surface by discretizing the possible ligand positions, orientations, and dynamics. While the LES provides the most accurate accessibility for a specific ligand, its computation takes several minutes for mid-sized proteins and a reasonable surface quality. Thus, if interactivity is required, the SES is favorable. The LES should be favored if a more detailed view of a static molecule is needed.

**Convolution Surface Models.** Blinn [Bli82] introduced implicit modeling as an approximation of the molecular surface in 1982. He proposed the use of a Gaussian convolution kernel (see Figure 3) in order to blend atom potentials to represent the electron density function. The resulting surface is commonly known as *Metaballs*, *blobby surfaces*, or *convolution surfaces* [VFG98]. Such a summation-based model, however, generally lacks information of the associated solvent molecule. Therefore, Grant and Pickup [GP95] determined the parameters for the Gaussian-based model to mimic the volume and solvent accessible surface area for different solvent probe sizes.

There are several other kernels mentioned in the literature

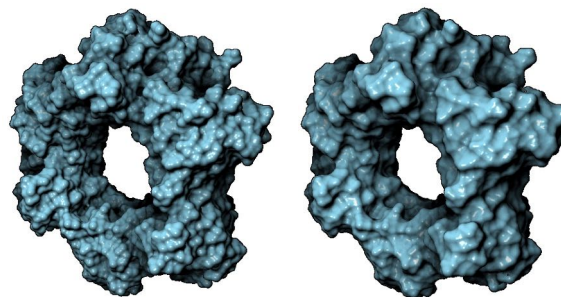


Figure 4: Molecular surface for proliferative cell nuclear antigen (PDB ID: 4D2G) represented by an implicit model defined by blending scheme [PB13]. The images show the result for different solvent radii: 1.4 Å (left) and 2.2 Å (right).

that can be used as alternative kernel functions [She99], i.e., avoiding computationally expensive exponential functions. One of the main advantages of kernel-based models is the simplicity of the representation and model evaluation. For instance, the function to be evaluated has linear time complexity and the final formula can be expressed analytically. In 2013, Parulek and Brambilla [PB13] proposed another implicit model with linear complexity although its definition is not purely analytical compared, e.g., to the Gaussian model. On the other hand, it resembles the SES more closely than the kernel-based approaches (Figure 4). The main reason lies in the fact that the implicit function evaluation incorporates the solvent, represented by a sphere of a specific radius. An implicit space mapping is then exploited to approximate the circular distance to individual atoms.

In 2008, Kanamori et al. [KSN08] proposed an efficient technique for ray casting the kernel-based models. It employs Bezier clipping to quickly compute an intersection between a ray and the surface. The GPU implementation exploits depth peeling to retrieve contributing spheres for the actual ray segment, where the iso-surface point is then evaluated through the Bezier clipping technique. To further speed-up the algorithm, Szecsi and Illes [SI12] suggested to employ *fragment linked lists* or an *A-Buffer* to avoid the multi-pass rendering required by depth peeling.

In order to visualize models based on implicits, they are often discretized on a regular grid prior to rendering. Subsequently, a triangle mesh can be extracted for rendering, e.g., using Marching Cubes. However, when dealing with complex shapes such as molecular surfaces, a very fine-grained tessellation is needed for a fully detailed surface representation. To remove this limitation, Krone et al. [KSES12] proposed an interactive visualization method to extract and render a triangulated molecular surface based on Gaussian kernels. They efficiently exploited GPGPU capabilities to discretize the density field, which is then processed by a GPU-accelerated Marching Cubes algorithm. The rendering performance depends on the resolution of the density grid as

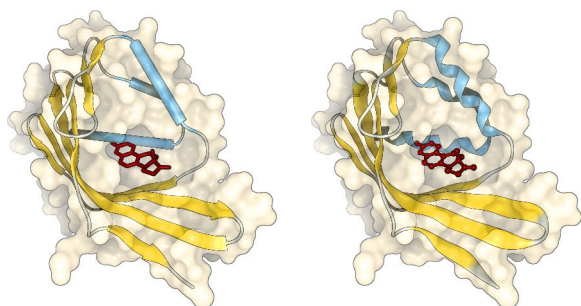


Figure 5: Two possible cartoon renderings of the same protein (PDB ID: 1OGZ). Left: Ribbon-shaped arrows show the direction of the amino acid chain for the  $\beta$ -sheets while the  $\alpha$ -helices are stylized as cylinders. Right: Rounded ribbons are used to illustrate sheets and helices. The semi-transparent SES is shown for reference. The ligand equilenine (red) is visualized in ball-and-stick representation.

well as on the number of atoms. Their method achieves interactive frame rates even for molecules counting millions of atoms due to the high degree of parallelism and is currently among the fastest molecular surface extraction algorithms.

## 4.2. Illustrative and Abstract Models

Apart from molecular models that show a direct representation of the atoms of a molecule, several abstract models have been established. An abstract model might illustrate a special feature of the molecule, which is not or at least not clearly and easily discernible in an atomistic model. Abstract models can also lead to sparse representations, which might be easier to understand or reduce occlusion. Such a representation can be useful, e.g., for a very large molecular complex, for which often not the individual atoms but the overall shape are of interest.

### 4.2.1. Representations of Molecular Architecture

Very early on, the conceptualization of complex macromolecular assemblies motivated scientists to simplify computer graphics images representing these entities. Visual abstraction of the molecular architecture often shows important structural features more clearly than a full-detail atomistic representation [MM04], e.g., using abstractions for molecular subunit structures [NCS85]. Goddard and Ferrin alternatively refer to such abstractions as multiple levels of detail that match the underlying structural hierarchy of molecular assemblies [GF07]. As our understanding of biological structures progresses, the need for new abstractions may arise such as it was the case for representing the bases of nucleic acid polymers and, more recently, carbohydrates.

In 1981, Richardson [Ric81] introduced the *cartoon* representation for proteins, which depicts the secondary structure as *ribbons* and *arrows*. Since then, a variety of car-

toon renderings have been developed that vary the graphical appearance, e.g., using straight cylinders for helices (see Figure 5). One of the earliest implementations of the cartoon model was the *Ribbons* algorithm by Carson and Bugg [CB86], which was influential to subsequent work. Its successor *Ribbons 2.0* [Car91] provided interactive visualization. A current challenge is to improve the efficiency for the interactive visualization of large, dynamic proteins. This can for example be achieved by mesh-refinement techniques at the software level [HOF04] or by moving to GPUs at the hardware level. TexMol by Bajaj et al. [BDST04] implements helix ray casting by using impostor-based GPU shaders instead of tessellated geometry. Several GPU implementations that generate the geometry on the fly were proposed, starting with Krone et al. [KBE08] comparing CPU, hybrid CPU/GPU, and full GPU implementations that exploit the geometry shader. Although with the available graphics hardware at that time the best performance was achieved with the CPU implementation, this might be no longer the case due to recent GPU developments. Using a hybrid CPU/GPU approach that uses only vertex shaders, Wahle and Birmanns [WB11] report a near 3-fold speed-up for their cartoon implementation. Recently, Hermosilla et al. [HGVV15] used tessellation shaders to further speed up the on-the-fly generation of the cartoon model. New variants of helix abstractions, with the aim to map simulation analysis data onto them, were proposed by Dahl et al. [DCS12].

Vehlow et al. [VPL11] presented an application that shows contact maps of the amino acids within a protein together with a three-dimensional representation of the protein. This allows the user to analyze the protein structure and compare amino acid contacts of different folds of a protein. The visualization was inspired by Ramachandran plots [RRS63], which show the backbone torsion angles of a protein. These plots are used to identify secondary structure elements (e.g., helices or sheets) of proteins and as an indicator for the quality of experimentally derived structures.

Abstracted representations are also used for DNA and RNA. DNA is commonly depicted by a ladder-like double helix representing the sugar backbone by a ribbon or tube and the nucleotide bases by sticks or ellipsoids. Many visualization tools feature such depictions, e.g., *VMD* [HDS96], *PyMOL* [DeL02], or *Chimera* [CHF06]. Ellipsoids are also used as a generic abstraction shape for a variety of structural elements in diverse classes of molecules [GMG08, AP09]. *RiboVision* by Bernier et al. [BPW\*14] is a specialized visualization tool for the structure of the RNA in ribosomes. It uses a combination of 1D plots, 2D sequence diagrams, and 3D visualization using linked views. This allows users a comprehensive analysis of the structure of RNA molecules.

Although glycoscience is an active field of research, there are only few abstracted representations tailored to carbohydrate molecules. Some simple geometric abstractions of

the atomic ring structures have been developed over the last decade, e.g., [CKSG09, PTIB14].

#### 4.2.2. Surface Abstractions

Molecular surface abstractions are typically based on the established molecular surface models detailed in Section 4.1.2. As explained in Section 2, biological macromolecules like proteins and DNA or RNA are composed of small molecular building blocks, namely amino acids in case of proteins and nucleotides in case of DNA or RNA. In a simple abstraction of the vdW surface one represents these building blocks by one or more tight-fitting bounding spheres that contain the individual atoms. In case of a protein, this simplification reduces the number of spheres on average by an order of magnitude, while maintaining the general shape of the protein. Similar simplifications are also used in coarse-grained molecular simulations to reduce the complexity and computation time [Toz05, Cle08]. This abstraction is for example available in the molecular visualization software VMD [HDS96] as *Beads* representation. Since the resulting surface abstraction consists of spheres, fast GPU-based ray casting can be used for rendering.

The convolution surfaces mentioned above can be used to obtain a smooth surface abstraction if correct parameter values are chosen. A larger kernel function in combination with a higher iso-value for the surface extraction results in a smoother surface that shows the general shape of a molecule instead of individual atoms. Such smoothed surfaces are especially useful for large molecular complexes consisting of up to several millions of atoms like virus capsids [KSES12].

Cipriano and Gleicher [CG07] presented a surface abstraction technique based on a triangulation of the SES. It smoothens surface parts that have low frequency and are, therefore, deemed less important while maintaining salient surface features. Textures can be used to highlight removed surface features such as bumps or indentations as well as binding sites for ligands.

Several techniques that map a molecular surface mesh (typically a triangulated SES) to a spherical coordinate system have been proposed. Rahi and Sharp [RS07] developed a method that uses a parametrization based on spherical coordinates to map the triangles of a molecular surface onto a sphere. The technique of Postarnakevich and Singh [PS09] uses a force-directed approach to deform a bounding sphere until it matches the SES, thereby creating a mapping between the SES and the sphere. Using this mapping, the sphere can be colored according to physico-chemical properties of the molecule or according to the path length of the sphere deformation to highlight the shape of the original SES. Hass and Koehl [HK14] use a conformal mapping between the molecular surface and a bounding sphere to measure how spherical the molecule is. They also propose to use the intermediate spherical representation to compare molecules. Of these three approaches, the method by Postar-

nakevich and Singh is the only one that can be used for molecular surfaces of genus 1 or higher; the other two methods are applicable only to molecular surfaces of genus 0.

#### 4.3. Structural Level of Detail

Molecular visualization often aims to render large molecular structures and systems in real time. However, at a certain size of molecular data it becomes even difficult to visualize simple models, like the vdW surface. Since displays are restricted in the number of pixels, in scenes with many million atoms, most atoms are either not inside the view frustum, occluded, or so distant to the camera that their projection is significantly smaller than a pixel. Level of detail (LOD) strategies can be applied to handle such problems. On the one hand, LOD methods can be semantic, that is, show an abstract version of the molecular structure; such approaches are especially useful to reduce clutter. On the other hand, LOD methods are often used to enhance the rendering performance, e.g., by detecting elements in the scene that are occluded by others or by using low-detail proxies for distant objects. Most existing methods present a seamless visual abstraction, incorporating different levels of abstraction into one molecular model.

When focusing on the semantics, molecular systems may be visualized with various degrees of structural abstraction, i.e., different parts of the system are rendered using different representations. Van der Zwan et al. [vdZLBI11] described a GPU implementation for visualizing continuous transitions between vdW surface, ball-and-stick, and cartoon model. They also proposed methods to support spatial perception and enhance illustrativeness (cf. Section 5).

On the other hand, there are several solutions that focus on the spatial arrangement of molecules. Bajaj et al. [BDST04] presented a biochemically sensitive LOD hierarchy for molecular representations. Their hierarchical image-based rendering also allows mapping of dynamically computed physical properties onto molecular surfaces.

Later, Lee et al. [LPK06] introduced an algorithm for view-dependent real-time surface rendering of large-scale molecular models. Their approach combines an adaptive LOD visualization of the molecular model with a high quality rendering of the active site. It is based on a two-step view-dependent method: In a pre-processing stage, the mesh representing the molecular surface is simplified and classified to different LODs; in a real-time rendering stage, hierarchical LOD models which are stored in a bounding tree are constructed to increase the performance.

Convolution surfaces like the fast molecular surface extraction by Krone et al. [KSES12] can also be used for LOD renderings. As mentioned in Section 4.2.2, this approach is able to display the structural detail on a continuous scale, ranging from atomic detail to reduced detail visual representations based on the chosen grid resolution and density

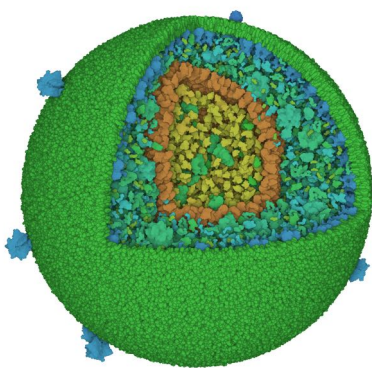


Figure 6: Illustrative visualization of a HIV virus generated by cellVIEW [MAPV15].

kernel function. Furthermore, groups of adjacent particles can be replaced by their bounding spheres, similar to coarse-graining. If these spheres are used as an input for the convolution surface calculation, the resulting surface approximates the original shape with reduced detail.

There are a couple of methods that focus in the GPU-accelerated rendering of partly rigid structures. These methods essentially create an inverse LOD: the input data are only molecular positions from which an all-atom representation is reconstructed. In 2007, Lampe et al. [LVRH07] proposed two-level approach to visualize large, dynamic protein complexes. In the first level, each residue is reduced to a single vertex based on its rigid transformation. In the second level, the geometry shader reconstructs the atoms of the residue based on the position and orientation. The atom spheres are ray-casted in the fragment shader. An additional feature is the fish-eye distortion, which allows the user to get a better view inside the protein. This approach results in a 3-fold rendering speedup; however, internal transformations of the residues are not possible. In order to minimize the data transfer to the GPU, Le Muzic et al. [LMPSV14] extended this approach by storing the atom positions of a whole molecule in a texture. Each instance of the molecule is then formed just by a single vertex, where the atom positions are reconstructed using the tessellation and geometry shader. Furthermore, an LOD approach is applied, which linearly summarizes adjacent atoms into a single sphere depending on the distance to the camera. In contrast to Lampe et al. [LVRH07], this LOD approach is not restricted to protein data. Later on, Le Muzic et al. [MAPV15] presented a system, *cellVIEW*, to interactively visualize large molecular datasets (see Figure 6). The exploited techniques further advanced the performance of atomistic visualization by means of a real-time LOD selection technique implemented in the tessellation shader. The proposed approach allows to render datasets containing 15 billions of atoms at 60 fps. In addition,

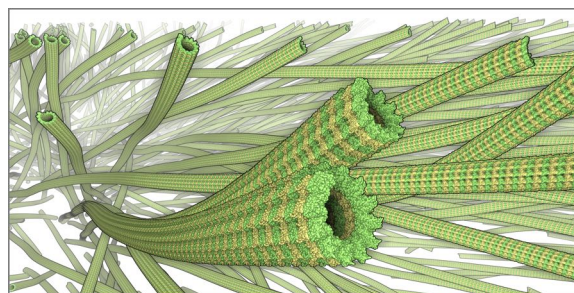


Figure 7: Microtubules reconstructed from electron tomography data and visualized as vdW surface using the approach by Lindow et al. [LBH12] with at least 3 fps on an NVIDIA GeForce GTX 470. The data set contains 4025 microtubules consisting of approximately 10 billion atoms.

tion, *cellView* was implemented in Unity3D game engine, allowing users to easily customize the system according to their needs.

In 2012, Lindow et al. [LBH12] presented an approach similar to those presented by Le Muzic et al. [LMPSV14, MAPV15], where the atomic data is stored in a 3D voxel grid on the GPU. During ray casting, fast ray-voxel traversal [AW87] is used and only spheres in the current voxel are tested for intersection. For large data sets, the rendering is much faster than direct ray casting [SWBG06] or even the two-stage culling approach by Grottel et al. [GRDE10]. Furthermore, the method exploits the fact that most biological structures, like microtubules and actin filaments, consist of recurring substructures. Hence, only one grid is created for each substructure of which many instances can be rendered with different rigid transformations. This approach can be used to interactively visualize biological scenes on atomic detail bridging five orders of magnitude in length scale with billions of atoms (see Figure 7). Shortly after, Falk et al. [FKE13] extended the technique by a hierarchical LOD to accelerate the rendering: if the projection of a grid cell is smaller than a pixel, it is not necessary to perform a ray casting with the spheres in a cell. It is only checked if the cell is empty or not. The same applies when the whole grid becomes smaller than one pixel. They also split the scene into several rendering passes. In each pass, the depth buffer of the previous pass is used for a depth test to avoid unnecessary ray casting operations. Furthermore, they presented a generalization of the approach for instances of triangulated objects. This enables the user to visualize complex models, like molecular surfaces.

Another view-dependent abstraction was proposed by Arndt et al., which is implemented in the GENOME tool [AAZ\*11]. They use different simple geometric abstractions to reduce detail in order to visualize the whole human genome. The simplified geometry makes it easier to identify particular components like histone proteins in an overview.

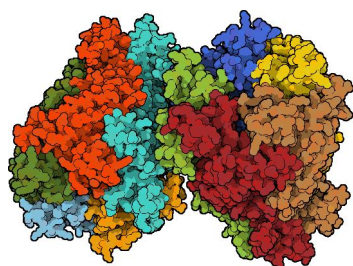


Figure 8: Non-photorealistic rendering of two proteins (PDB ID: 4A97) resembling the style used by Goodsell [Goo] for his Molecule-of-the-Month. Image made with MegaMol [GKM\*15].

Parulek et al. [PJR\*14] introduced a LOD method for fast rendering of molecular surfaces. Their method combines three molecular surface representations—SES, Gaussian convolution surface, and vdW surface—using linear interpolation (see Figure 1). The choice of the respective model is driven by an importance function that classifies the scene into three fields, depending on the distance from the camera. The hierarchical abstraction incorporates a customized shading that further emphasizes the LOD. The A-buffer technique is used to improve the performance.

## 5. Molecular Rendering

The visualization of molecular dynamics data is often crowded and features a high visual complexity besides a high depth complexity. Advanced real-time rendering and shading methods cannot only enhance the image quality but also enhance the perception of geometric shapes and depth complexity in the scene. The main aspects related to molecular visualization are shading and various depth cues including ambient occlusion effects. The most commonly applied techniques in this context are discussed in the following. All methods listed below have in common that they can be computed for dynamic data in real-time.

The *color* of the rendered representations is usually obtained from the type of the atoms, chains, functional units, bonds, or other derived attributes. The oldest and most simple coloring method is to assign individual colors to the chemical elements. Biochemical properties of the molecules are usually color-coded onto the atoms.

Other properties that can be mapped onto all types of molecular models using per-atom coloring include for example B-factor, flexibility, hydrophobicity, amino acid chain, or partial charge. The prevalent shading models used for *illumination* in molecular visualization are Phong [Pho75] and Blinn-Phong [Bli77]. However, specular highlights created with both models tend to create artifacts due to high frequencies. Grottel et al. [GRDE10] proposed a normal correction scheme to smooth out these high frequencies between adja-

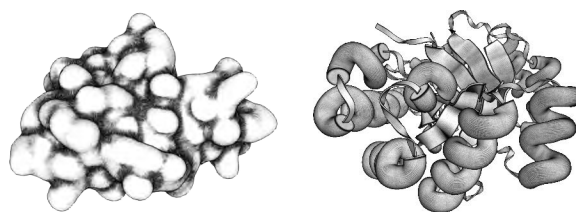


Figure 9: Illustrative line renderings of two molecules: surface structure (left, image source: [LKEP14]; PDB ID: 1OGZ) and cartoon representation (right, made with ProteinShader [Web09]; PDB ID: 1RWE).

cent normals of distant objects. This normal correction results in a more continuous lighting that creates surface-like impressions for distant molecules [GRDE10, LBH12].

Inspired by hand-drawn illustrations of the molecular interior of cells done by David Goodsell [Goo09, Goo], *toon shading* is often used to produce artistic or non-photorealistic renderings with a comic-like look. In Figure 8, this type of shading is applied to the protein B-Raf.

Illustrative representations using *line drawings* consisting of feature lines and hatching have a long tradition in molecular rendering. See [RCDF08] for an overview on line drawings. In particular, contour lines are widely applied in molecular visualization [TCM06, LVRH07, KBE09]. Goodsell and Olson use several types of hatching to illustrate molecular surfaces [GO92]. Contour lines and hatching have also been applied to yield a continuous abstraction between an atomistic model and a cartoon model of a protein [vdZLB11]. The ProteinShader tool by Weber [Web09] offers line-based real-time illustrative rendering for cartoon representations of proteins. Lawonn et al. [LKEP14] combined feature lines and hatching to emphasize important features on molecular surfaces. The method is based on line integral convolution (LIC) on the vector field of the illumination gradient, which emphasizes salient surface regions. Figure 9 shows examples for illustrative visualizations of proteins.

*Ambient Occlusion* (AO) is a method based on the works of Miller [Mil94] and Zhukov et al. [ZIK98] that mimics the transport of diffuse light between objects leading to localized shadowing in creases, which can increase depth perception. AO works best for dense particle data sets, which makes it suitable for most molecular data visualizations [TCM06]. Since AO is computationally expensive, several accelerated approaches have been developed for interactive visualization. Screen-Space AO (SSAO) is an image-space technique that approximates the effects of AO in a postprocessing step, e.g., [Kaj09]. For molecular data sets, Object-Space AO (OSAO) techniques can yield even more convincing results. OSAO considers the entire local neighborhood, unlike SSAO approaches that can only consider the visible neighborhood. Grottel et al. [GKSE12] developed an OSAO method that reaches interactive frame rates even for very

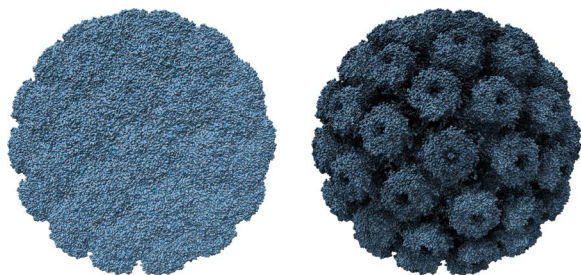


Figure 10: Rendering of a virus capsid (PDB ID: 1SVA) with local illumination (left) and ambient occlusion (right). Unlike the local lighting, the ambient occlusion highlights the capsid structure clearly (made with MegaMol [GKM\*15]).

large, dynamic particle data sets. The method uses a volumetric approximation of the local neighborhood to store the ambient occlusion factors. Recently, this approach was extended by Staib et al. [SGG15] using a hierarchical voxel-cone tracing method improving the sampling of a full-color AO map. Their method also works for transparent particles. To avoid computing the AO map, Eichelbaum et al. [ESH13] presented PointAO, a screen-space method for particle rendering that focuses on retaining both global and local structural information, which is an extension of their previously presented technique LineAO [EHS13]. In Figure 10, the differences between local illumination and OSAO are shown. Figure 3 depicts a combination of depth cueing, silhouettes, and SSAO for molecular surfaces. Note that the abovementioned interactive AO approaches are only the most widely used ones for molecular visualization, as a comprehensive list of AO methods would be out of scope of this report.

Distinct object boundaries are a beneficial *depth cue* for scenes with many objects, like proteins or simulation results. Depth-dependent *silhouettes* [ST90] can be computed in image space in a post-processing step by detecting discontinuities in depth and adjusting line widths accordingly. A similar effect is obtained by applying *halos* extending from the object boundaries as proposed by Tarini et al. [TCM06]. At the boundary of the object, the halo features the same depth as the object. With increasing distance from the object, the depth of the halo increases as well. A similar technique, the *depth darkening* approach by Luft et al. [LCD06], separates distant overlapping objects visually and creates depth-dependent halos in image space. Simple fogging or depth-dependent desaturation can be used as additional depth cues.

To separate features in the foreground from the background, the *Depth of Field* (DoF) effect from photography can be used where only the objects in focus are retained sharp whereas everything else appears blurred. In molecular visualization, DoF can be used to draw the attention of the user to a specific region and is computed interactively in image space [FKE13]. Kottraval et al. [KFSR15] recently proposed an object-space approach for DoF utiliz-

ing a coverage-based opacity estimation which can be computed at interactive frame rates. The DoF effect can also be adjusted to highlight semantic properties [KMH01] like single bonds or charge densities within a protein.

Typically, the viewpoint and camera parameters are chosen by the user when rendering and exploring molecular scenes. The automatic choice of the best view for a particular molecule requires additional information besides the structural data to map the 3D structure onto the screen. Vazquez et al. [VFSL02] utilize the concept of viewpoint entropy and extend it to orthographic molecular views. Incorporating additional semantic information on the protein can improve the selection of an optimal camera setting [DCMP10].

## 6. Visualization of Molecular Dynamics

As mentioned in Section 2, molecular simulation is nowadays an important source of data. Simulations can compute the individual trajectories of all atoms over a certain time frame. The resulting time-dependent data can provide insight into the dynamics of the simulated molecular system on an atomistic level. Note that in this context *molecular dynamics* does not specifically refer to the results of a MD simulation, but to time-dependent molecular data that represents the dynamic behavior of the molecules.

The molecular models discussed in Section 4 can naturally be used to visualize dynamic data. They represent the instantaneous conformation of a molecule for a given snapshot and can show how it changes over time using animation. In this section, molecular visualizations are discussed that go beyond these basic models by extracting and visualizing the abovementioned dynamic behavior of the molecule. Several resources for such dynamic data exist [McG08, Iwa08, JH14, Ber07, KG00] and provide for instance short movies describing molecular functions based on their structure. Since these solutions are created for educational purposes, they mainly focus on the artistic appearance and use pre-rendered, non-interactive visualizations.

### 6.1. Visualization of Flexibility

Molecules are intrinsically flexible entities, yet the vast majority of visualizations represent a static structural snapshot. To account for the positional uncertainty, precisely defined atomic positions may be replaced by probability distributions to depict varying molecular conformations [RJ99]. Representations for dynamic molecular conformations were further investigated by Schmidt-Ehrenberg et al. [SEBH02]. They developed a method to sample ball-and-stick and vdW representations onto a grid including color to depict atomic or residual properties. The conformational fuzziness thus computed is then shown using isosurface or direct volume rendering. *MolMol* [KBW96] and several other programs provide “sausage” views that are similar to this method,

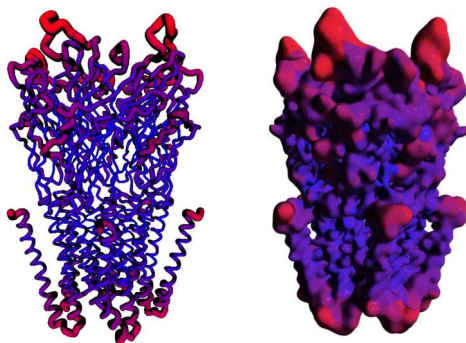


Figure 11: Representation of the backbone flexibility of the GLIC ion channel protein (PDB ID: 4HFI) by a tube of varying radius (left) or a flexibility isosurface (right). Flexible regions (red) occupy more space than well-defined rigid parts of the molecule (blue). Image generated with Unity-Mol [LTDS\*13].

where abstracted representations such as a protein backbone tube are modulated according to a pre-calculated flexibility parameter (see Figure 11). The width of the resulting tube highlights the flexibility. Lee and Varshney [LV02] depicted thermal vibrations of atoms through multi-layered semi-transparent surfaces. Selected flexible elements such as loops or domains in proteins can be represented by voxel maps [CBES11]. Bryden et al. [BPG12] used glyphs to illustrate molecular flexibility calculated from normal mode analysis. Their approach clusters groups of atoms that exhibit a synchronized rotational motion. The clusters are highlighted and equipped with the corresponding circular arcs that illustrate the rotation. Arrows on top of these arcs show the direction of the rotation and other values like velocity, error, or nonrigid energy. Fioravante et al. [FSTR13] presented visualization methods to analyze motional correlations in proteins. These correlated motions are identified based on principal component analysis and covariance clustering. The results of these analyses are used to enrich the 3D visualization of the protein structure, e.g., using color or cone glyphs.

Heinrich et al. [HKOW14] presented a visual analysis application tailored to intrinsically disordered proteins. Such proteins have very flexible regions that can exhibit a wide range of three-dimensional structures depending on external factors [UD10]. The application shows a 3D visualization of an ensemble of superimposed structures as well as a parallel coordinates plot [Ins09] with per-residue statistics. This plot can be used to filter or cluster the protein structures and to find correlations between them.

Recently, Dabdoub et al. [DRSR15] presented the tool *MoFlow* that visualizes the dynamics of a molecule by rendering the pathlines of selected atoms of the molecular structure, e.g., backbone atoms. The atom positions between time steps are interpolated using splines. The resulting curves are

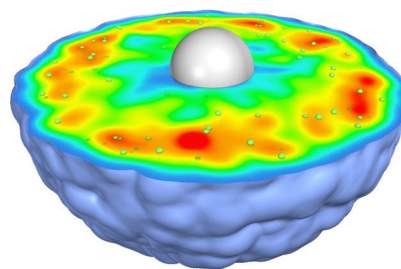


Figure 12: Visualizing cellular signaling processes with a volumetric representation obtained from discrete signal proteins [FKRE10]. Red indicates a high concentration of signal proteins whereas blue indicates very low values.

colored according to a timescale color map allowing an easy understanding of the movements of the atoms over time. More visual cues are added through semi-transparent ribbons displaying the movement of bonds. While *MoFlow* allows an easy understanding of short parts of a trajectory, the visual representation might quickly get confusing for very complex movements.

## 6.2. Volumetric Representations and Aggregation

Besides the specialized molecular representations discussed in Section 4, visualization methods developed for other application fields can also be utilized to depict molecular data sets. Especially vector field visualization methods can be useful for dynamic molecular data. These methods, however, require a continuous representation of the raw particle data. Such a representation can be obtained by sampling points to a 3D grid. Similar to the convolution surfaces, a kernel function is often used to define the influence radius of the sampled particles. Cohen et al. [CKK\*05] used volumetric maps to study accessibility. Scharnowski et al. [SKS\*13] sampled dipole moments derived from the atomic positions to a grid and subsequently used the curl operator to separate similar regions in the resulting vector field. They rendered isosurfaces around these consistent regions. The isosurfaces were textured using line integral convolution in order to show the directions of the dipole moments. Falk et al. [FKRE10] sampled the positions of signal proteins in whole-cell simulations to a grid in order to show the development of the signal density using direct volume rendering (see Figure 12).

Aggregation is a commonly used concept to reduce the dimensionality of scientific data. Rozmanov et al. [RBT14] sampled atoms with different properties to separate grids in order to obtain spatial atomic densities. They also aggregated several time steps into the grid by averaging the local property values of the atoms. The aggregated densities are also visualized using isosurfaces. Temporal aggregation of atom densities and their properties was also used by Thomaß et al. [TWK\*11] to visualize the average probability of pres-

ence for the components of a mixed solvent around a hydrogel. The results are color-mapped to an averaged molecular surface of the hydrogel. An alternative representation is to use volume rendering as did Durrieu et al. [DLB08] to illustrate water occupancy around a protein averaged over a MD simulation. In specific cases, such as when a cylindrical geometry around a protein channel is observed, the dimensionality of the representation can be further reduced to map, e.g., the solvent density in 2D as in [BS03]. Chavent et al. [CRG\*14] aggregated the diffusional motion of lipids on a grid and visualized the diffusion using arrow glyphs and streamlines. A similar approach was used by Ertl et al. [EKK\*14] to analyze the motion of ions around DNA in a nanopore. Due to the repetitive nature of the DNA and the periodic boundary conditions, they not only used temporal but also spatial aggregation of the ion densities and velocities. They combined different visualization methods for the analysis of the data (pathlines, isosurface, LIC, glyphs). A key point for most temporal aggregation methods is that the center of mass does not change significantly during the time frame of interest. Depending on the simulation, this might be given implicitly (see, e.g., [EKK\*14]). In other cases, a central molecule that moves freely during the simulation has to be aligned onto a reference frame. For molecular data, alignment by RMSD minimization [Kab78] is commonly used to superimpose the time steps.

Aggregation and clustering has not only been applied to grid-based molecular data. Lindow et al. [LBBH12] for example used aggregation to illustrate time-dependent channels of proteins. Their method extracts the channel state for each time step and aggregates partially open channels if they are connected in consecutive time steps to show in one static image whether they can be traversed over time. Byška et al. [BJG\*15] used an aggregated bottleneck contour (collar) to focus on a given channel constriction. Bidmon et al. [BGB\*08] aggregated solvent pathlines—i.e., the trajectories of individual solvent molecules—to investigate the water movement near protein cavities.

### 6.3. Interactive Visualization and Manipulation of Molecular Models

Visualization is an essential element of interactive simulations. As the visualization has to be interactive for the user to be able to steer the simulation properly, simulation performance typically is the main limiting factor. Interactivity has been a target for molecular graphics since the 1960s [Fra02]. At that time, interaction meant essentially controlling camera movement. The element of active manipulation was added later on, first by a specialized energy minimization approach, starting with 20 to 80 residues systems, and eventually leaving out electrostatic interactions [SRRB94]. *MDScope* did interactive visualization and steering for MD simulations with full electrostatics up to a few hundred residues, and raised the issue of timescale limi-

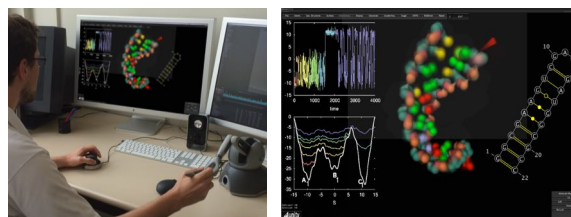


Figure 13: A typical steered simulation workspace. The user moves the camera with his left hand while manipulating the molecule with the haptic device in his right hand (Geomagic Touch, <http://www.geomagic.com/>). The red cone represents the virtual probe that can be used to select atoms and apply forces using the haptic device. Simulation-related information can be overlaid, e.g., plots of distances or energies as well as additional representations such as the 2D view of the simulated RNA molecule to the right side of the screen. Visualized within UnityMol connected to a simulation running in the *HireRNA* software [CDB\*15].

tations [NHK\*95]. Especially in the context of steered simulations, haptic feedback using specialized interaction devices becomes interesting since it can be used to convey forces. Another application area that uses methods for direct manipulation of molecular data is interactive molecular modeling. Intuitive haptic exploration using specialized hardware was implemented [SGSG01] and applied to a 4000 atom membrane channel. The performance requirement for haptic rendering is even more stringent than for graphics rendering, as it imposes refresh rates of about 1000 Hz. The state-of-the-art methods for atomistic molecular visualization detailed in Section 4 are able to handle dynamic data in real time. Thus, they can be used for visualizing interactive simulations, in particular with simplified—so-called coarse-grained—representations that can be calculated fast enough; a typical setup is depicted in Figure 13. Nowadays, with such approaches, even chemical reactivity can be explored interactively at the quantum level [HVB\*14]. Such interactive experiments are facilitated by visual manipulation guides discussed by Kreylos et al. [KMH\*03]. They developed the molecular modeling tool *ProteinShop* that uses inverse kinematics for the constraints within a molecule, which helps users to determine optimal protein folds [CKH\*04]. Nowadays, with cheaper hardware, better graphics cards, and faster computers, haptic steering has become very attractive [SKVS10] and can be applied even to systems comprising more than one million atoms [DPT\*13].

### 6.4. Visualization of Molecular Reactions

Understanding molecular interactions in living organisms is essential to understand their physiology and is often a basis for drug design in pharmaceutical research. Modeling of coupled molecular reactions is, thus, one of the research foci in systems biology. The most widely used



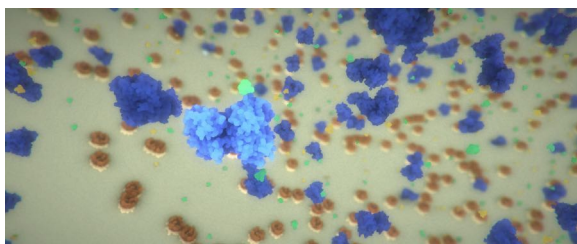


Figure 14: Screen capture of the illustrative timelapse method presented by Le Muzic et al. [MWPV15]. It highlights a succinate thiokinase (light blue) and ATP (green) produced by the reaction of the thiokinase with ADP.

tools include *CellDesigner* [FMKT03], *VCell* [MSS\*08], *TinkerCell* [CBS09], *BioNetCAD* [RFD\*10], *Rulebender* [SXS\*11], *NetworkViewer* [CAZMS14], and *CytoScape* [SMO\*03]. Besides visualizing the quantitative change of reactants in time-intensity curve plots, these tools offer various network visualizations. These range from following the Systems Biology Graphical Notation [LN\*09] to illustrative textbook-like depictions of the modeled processes. However, the visualization of kinetic models primarily focuses on relational and quantitative aspects, actual behavior of involved reactants is not communicated.

Falk et al. [FKRE09] propose several methods to visually emphasize interesting aspects of particle-based cellular simulations like particle trajectories. *MCell* simulations can be visually inspected by *CellBlender* [BDF15], which is a plugin for the 3D modeling tool *Blender*. The visualization module eases generation of *MCell* models and shows the resulting simulation, where the molecules, represented as glyphs, are embedded into 3D meshes of cellular structures. *ZigCell3D* [dHCKMK13] is another system for designing and visualizing cellular models. It offers a visualization on the atomistic level while visually highlighting reactions between particles. Since such particle simulations are typically very crowded, interactions might still be missed. Thus, Le Muzic et al. [LMPSV14] proposed a technique to visually represent a particle-based system with an underlying quantitative simulation. This simulation is steered by the visualization so that reactions happen in front of the user to convey the spatial aspects of the reaction chain. They later extended their technique with a specialized illustrative time-lapse method (see Figure 14) that slows down the movement of proteins while they are involved in a reaction [MWPV15].

Tek et al. [TCB\*12] provided an environment to model and visualize protein-protein interactions. Visual cues can be complemented by multi-modal audio and haptic feedback, thus ‘rendering’ interactions calculated from live molecular simulations on multiple sensory channels.

Particle-based models have also been employed in visualization of polymerization where reactions add build-

ing blocks onto existing polymers [GIL\*10]. Kolesár et al. [KPV\*14] use a multiscale particle model for illustrating polymerization where the system parameters can be tweaked interactively. Thus, the user receives an instantaneous visual feedback on the growth process of the polymer.

## 6.5. Visualization of Quantum Effects

Understanding details of reactions requires quantum chemical studies, i.e., analysis of the electronic structure of molecules by computing the ground state, the excited states, and the transition states that occur during chemical reactions. For an elementary introduction see, e.g., [Heh03]. The resulting data are expectation values of physical variables, like, e.g., electron and nuclear densities or fluxes, describing, e.g., equilibrium geometries and reaction energetics.

On the visualization side the depiction of fields and multi-fields is required. It has been demonstrated that visualization of such fields helps to reveal rich and surprising phenomena (occurring even in simplest molecular systems) [BHI\*09, ABB\*11, HKM\*11]. Topological visual analysis of electron density fields provides information about the spatial domains attributed to individual atoms [Bad90]. There are a number of methods and tools to visually analyze covalent and non-covalent bonds [GBCG\*14], weak interactions [JKMS\*10, CGJK\*11], and molecular orbitals [SSH\*09] (see Figure 15) as well as related electron densities [HG08]. Also visualization of the resulting fuzzy molecular surfaces using volume rendering has been proposed [KCL\*13].

The understanding of photoelectron transfer processes in molecular systems also requires quantum mechanical approaches, like, e.g., time-dependent density functional theory. For a recent example presenting tools to visualize and analyze such process see, e.g., [GHZ\*15]. In photosynthetic systems typically electronic and vibrational degrees of freedom are coupled to transfer the energy between chlorophylls; in addition quantum mechanical entanglement and coherences between different parts of molecular complexes play a role for functioning in photosynthesis [KK12]. It is obvious that such complex spatio-temporal processes can be understood (and related to experimental results from multi-dimensional femtosecond spectroscopy) only with the help of advanced visualization techniques.

## 7. Molecular Visualization Systems

In this section, our aim is not to provide the readers with an exhaustive list of existing systems for molecular visualization, as such lists are emerging quite often in the literature. We rather present the most commonly used and robust systems incorporating most of the techniques presented above.

In the last decades, many tools and systems for molecular visualization have emerged. Some of them were designed for a specific purpose and their development has ceased. On

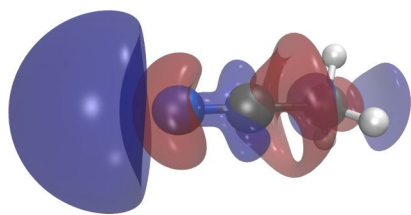


Figure 15: Illustration of molecular orbitals for an acetonitrile molecule (made with VMD [HDS96]).

the other hand, there are several very successful and robust systems that are commonly used by domain experts both for visual analysis in their research and for dissemination of results. We decided to categorize the existing systems to four groups: freely available functionally rich systems integrating some of the state-of-the-art methods, open-source prototype tools focused on efficient algorithms and extensibility, commercial systems, and web-based solutions. This section is structured with respect to this categorization.

The first category contains robust and popular tools, such as *VMD* [HDS96], *PyMOL* [DeL02], *Chimera* [PGH\*04], *YASARA View* [KV14], or *CAVER Analyst* [KSS\*14]. These systems are freely available for non-commercial purposes and, hence, widely used by the scientific community. Some of these systems also gain from the user community that contributes by adding own plug-ins. Most of the systems support all basic representations of molecular models discussed in Section 4. Many tools additionally provide means to equip the traditional molecular models with additional information about various physico-chemical properties and relationships in the molecular system (e.g., atomic densities, molecular orbitals, polarization, or electrostatic potentials and fields). Their proper visual representation can provide important insight into bonding and other relationships. The molecular orbitals (see Figure 15) can be computed and visualized for dynamic data using GPU-accelerated algorithms [SHLK11]. Tools like VMD, Chimera, and PyMOL furthermore enable users to load field data stored on regular grids, which can then be visualized by mesh extraction, iso-contours, or volume rendering. They also offer field line visualizations, which can be useful for electrostatics data. There is also a variety of specialized stand-alone tools for molecular visualization of such physico-chemical properties, such as *Molden* package [SN00], *Molekel* [PL00], *Gabedit* [All11], *GaussView* [DKM09], *Chemcraft* [And15], and *Avogadro* [HCL\*12]. All these tools, as well as VMD and Chimera, are also able to visualize molecular orbitals that are read from cube files which are output by tools like the *Gaussian* [FTS\*09] or *GAMESS-US* [SBB\*93] programs. At least Avogadro uses OpenGL shaders to efficiently create high-quality renderings of molecular orbitals.

The second group of systems is formed by single-purpose or prototype tools, which are also freely available and most

of them are open-source as well. The greatest advantage of such systems is that they focus on very efficient implementations with respect to latest advances in molecular visualization and rendering. One example is the *QuteMol* tool by Tarini et al. [TCM06], which was created to demonstrate the benefits of edge cueing and ambient occlusion. Another such tool is *ProteinShader* that showcases the illustrative cartoon rendering developed by Weber [Web09]. Other tools are released in the form of a prototype, sometimes as an open-source project that allows other developers to contribute. *MegaMol* by Grottel et al. [GKM\*15] is an open-source rapid prototyping framework that is tailored to molecular visualization. In order to enable the development of novel, efficient visualization methods, it is designed as a thin supporting layer on top of the OpenGL API. The modular framework allows developers to add extensions by implementing plugins. The underlying core library supports the developer with basic functionality but does not restrict in terms of data structures or technologies, which is the case for some special-purpose tools. Many of the aforementioned techniques were implemented using the *MegaMol* framework, e.g., GPU-based cartoon models [KBE08], molecular surfaces [KBE09, KGE11, KSES12], and accelerated rendering and shading methods [GRDE10, GKSE12]. *UnityMol* [LTDS\*13], another open-source prototype tool, was initially designed as a proof of concept to evaluate whether a game engine might enable domain scientists to easily develop and prototype novel visualizations. It was shown that a molecular viewer with original features such as animated field lines, lit spheres lighting, HyperBalls shaders [CVT\*11] and more could be implemented easily and quickly. The main drawback is limited performance due to the overhead of the game engine and the nature of molecular objects, which exhibit particular properties such as an increased number of triangles and required draw calls, compared to typical video game objects. The implementation of an extended graphical user interface is very costly, too. Recently, *UnityMol* has been extended to prototype visualizations of carbohydrate molecules [PTIB14] and to act as interface for interactive molecular simulations. Furthermore, *UnityMol* provides a free and open starting point for video game developers and scientists who want to use molecular objects in *Unity3D* projects. An example of a prototype tool based on a 3D modeling and animation software is *BioBlender* [ACZ\*12]. It is a multi-platform add-on for *Blender*, aiming at providing tools for the import and elaboration of biological molecules. *Molecular Maya* [mma13] is a software toolkit that extends the capabilities of the professional *Maya* tool from Autodesk. It enables to import, build, and animate molecular structures. One of the latest software tools designed to assemble large scale molecular models consisting of building blocks is called *cellPack* [JAAA\*15].

The third category of systems is formed by commercial solutions like *MolSoft ICM-Pro* [ATK94] or *Amira* [SWH05]. There are also several commercial ex-

tensions for YASARA [KV14]. These systems partially incorporate the abovementioned state-of-the-art techniques. Amira, for example, provides all the classical representations like ball-and-stick, space-filling, and cartoon representations. Furthermore, molecular surfaces like vdW surface, SAS, SES, and MSS can be rendered using GPU-based ray casting [LBPH10]. Amira also provides alignment and grid-based sampling tools to effectively visualize the flexibility of molecules using iso-surfaces or volume rendering. In general, however, it is often difficult to assess the commercial tools technically due to their closed source.

The last category contains web-based solutions for molecular rendering. Although such tools usually cannot integrate the latest state-of-the-art techniques covered by this report due to technical limitations, it is worthwhile to mention them because they are nowadays capable of interactive visualization of large molecular complexes. They can be embedded into web sites to provide specialized visualizations of entries in structural databases or results of structure-related calculations. One of the most traditionally used web-based tools is the Java-applet *Jmol* [jmo09]. It supports loading many file formats, rendering molecular surfaces, orbitals, schematic cartoons, and other features. *OpenAstexViewer* [Har02] is another Java-based program which aims to assist in structure-based drug design. It can be used both as applet and as a standalone application. Among other functions, it offers shaded molecular surfaces with transparency and property mapping. *JSmol* [HPR\*13], an extension of *Jmol* that uses only HTML5 and JavaScript instead of Java, is currently under development. *NGL viewer* [RH15] and *iview* [LLNW14] are examples for modern web-based tools utilizing WebGL, which enables hardware-accelerated rendering in the browser. They both support a variety of molecular representations. Another example of an object-oriented JavaScript library for visualizing molecular data is *3Dmol* [RK15]. It also supports standard representations of the molecule, including transparent molecular surfaces and visualization of orbitals. As mentioned above, the established web-based visualization tools rely on triangle-based rendering due to the limitations of web-based graphics. Recently, Mwalongo et al. [MKK\*14, MKB\*15] showed that WebGL enables GPU-based ray casting in the browser. Such technological advances will probably lead to more advanced web-based molecular visualizations in the near future.

## 8. Conclusion and Future Challenges

Molecular biology is a very diverse field, which implies that the molecular visualization is diverse as well. Therefore, a plethora of different representations—each of them having particular advantages and disadvantages—have been developed using a wide variety of visualization techniques. Consequently, there is not one best representation but rather many specialized ones, each one best suited for a specific task. One very prominent trend in the recent years has been

to use GPUs not only for rendering but also for accelerating the underlying computations [CLK\*11]. Programmable GPUs and multi-core CPUs have been a driving factor for parallelization of the algorithms in order to interactively visualize larger and dynamic molecular data originating from molecular simulations. At the same time, modern GPUs are powerful enough to render high-quality images at interactive frame rates. This allows domain experts to visually analyze increasingly large and complex molecular data.

The constant improvements in data acquisition technology and simulation methods provide a continuous challenge for the visualization of the derived, increasingly large molecular data sets in terms of particle numbers as well as time steps. Thus, the development of efficient visualization algorithms remains a promising direction for future work, including out-of-core methods for the visualization of very large data sets covering long time scales.

Since advances in hardware development nowadays rather increase the degree of parallelism than the clock speed, pushing the limits of parallel computing is an important issue. This includes the efficient exploitation of multi-core CPUs as well as GPUs and compute clusters. Since clusters are already widely used for molecular simulation, a tight coupling of simulation and visualization methods can alleviate the in situ analysis of large molecular systems.

Of equal importance are advances in the development of efficient simulation algorithms. Molecular simulations are becoming increasingly faster; new simulation techniques revealing the essential dynamics of molecular systems allow for interactive simulation steering. Visually steered molecular simulations will certainly become a game changer. They will enable structural biologists to focus on the most interesting aspects of complex molecular processes by interactively changing all kinds of parameters as well as initial and boundary conditions.

Another emerging trend is the use of interactive ray tracing for molecular graphics, which allows the user to get publication-quality images in real time. Sample tools that offer real-time ray tracing are *BallView* [MHLK05], which was one of the first tools to offer a real-time ray tracing on the CPU, and the current version of *VMD*, which includes a GPU-accelerated ray tracing engine [SVS13]. Recently, Knoll et al. [KWN\*14] presented a parallel interactive volume ray casting of radial basis functions on CPUs.

From a more general perspective, biomolecular visualization will have to handle three major challenges: depicting physical phenomena in more detail, improving the perceptual and cognitive efficiency of visualizations, as well as depicting longer trajectories of larger molecular systems. All this will increase the significance of visual insight methods.

Regarding the first topic, instead of just depicting the molecular dynamics on a purely phenomenological level, the physical and chemical *causes* for molecular events should

also be visualized—both on the classical and quantum mechanical level. As compute clusters and simulation methods are improved, the number of quantum mechanical degrees of freedom that can be dealt with will increase. Therefore, novel visualization methods for the depiction of quantum phenomena in dynamic molecular systems will be needed.

Techniques improving the depth perception for complex molecular structures have been investigated extensively already (see Section 5). However, there are still opportunities to augment current visual representations with additional cues (e.g., [SVGR15]). Visual clutter can be addressed by developing new illustrative visualization techniques, such as specialized cutaways, unfolding, or exploded views.

Regarding increased size of input data sets, one has to deal with two problems: First, dealing with ever longer molecular trajectories; for this, new techniques will be needed, similarly to those used in interactive video analysis and video processing. Second, dealing with larger molecular systems; for this, new visual representations of the data will be required and, in consequence, a complete visual language for biomolecular systems needs to be established. This includes abstractions that go far beyond the level of single molecules. Today atomistic representations are available for viruses; soon small bacterial organisms will be modeled in atomic detail. When zooming out from a molecule to see the entire structure at some point, all the molecules in the model create a salt-and-pepper noise pattern without any strong informative insight. Currently there is no abstraction mechanism that would meaningfully convey these levels, as for example cartoon representations do for secondary structures. Maybe for such large molecular complexes we will be soon witnessing investigation in a meaningful definition and visual representation for quinary structure and even beyond?

**Acknowledgments:** This work was supported through grants from the German Research Foundation (DFG) as part of SFB 716, the Senate Department for Economics, Technology and Research in Berlin, the Excellence Center at Linköping and Lund in Information Technology (ELLIIT), the French Agency for Research grants ExaViz (ANR-11-MONU-003) and Dynamo (ANR-11-LABX-0011), the Vienna Science and Technology Fund (WWTF) through project VRG11-010, the OeAD ICM through project CZ 17/2015, the Norway grants project NF-CZ07-MOP-2-086-2014, and the PhysiIllustration research project 218023 funded by the Norwegian Research Council.

## References

- [AABA10] ANDREWS S. S., ADDY N. J., BRENT R., ARKIN A. P.: Detailed simulations of cell biology with Smoldyn 2.1. *PLoS Comput. Biol.* 6, 3 (2010), e1000705. 4
- [AAZ\*11] ARNDT W., ASBURY T. M., ZHENG W. J., MITMAN M., TANG J.: Genome3D: A viewer-model framework for integrating and visualizing multi-scale epigenomic information within a three-dimensional genome. In *BIBM Workshops* (2011), pp. 936–938. 11
- [ABB\*11] ANDRAE D., BARTH I., BREDTMANN T., HEGE H.-C., MANZ J., MARQUARDT F., PAULUS B.: Electronic quantum fluxes during pericyclic reactions exemplified for the cope rearrangement of semibullvalene. *J. Phys. Chem. B* 115, 18 (2011), 5476–5483. 16
- [ACZ\*12] ANDREI R. M., CALLIERI M., ZINI M. F., LONI T., MARAZITI G., ZOPPE M.: Intuitive representation of surface properties of biomolecules using BioBlender. *BMC Bioinformatics* 13, Suppl 4 (2012), S16. 17
- [ADP08] ABBOTT D., DAVIES P. C., PATI A. K.: *Quantum aspects of life*. Imperial College Press, 2008. 2
- [AKM14] AL-KHALILI J., MCFADDEN J.: *Life on the Edge: The Coming of Age of Quantum Biology*. Random House, 2014. 2
- [All11] ALLOUCHE A.-R.: Gabedit - A graphical user interface for computational chemistry softwares. *J. Comput. Chem.* 32, 1 (2011), 174–182. 17
- [AM06] ADCOCK S. A., MCCAMMON J. A.: Molecular dynamics: survey of methods for simulating the activity of proteins. *Chem. Rev.* 106, 5 (2006), 1589–1615. 3
- [And15] ANDRIENKO G. A.: Chemcraft – graphical program for working with quantum chemistry results, 2015. Current version: 1.8, [www.chemcraftprog.com](http://www.chemcraftprog.com), online Oct 2015. 17
- [AP09] ABRAHAMSSON E., PLOTKIN S.: Biovec: A program for biomolecule visualization with ellipsoidal coarse-graining. *J. Mol. Graph. Model.* 28, 2 (2009), 140–145. 9
- [ATK94] ABAGYAN R., TOTROV M., KUZNETSOV D.: ICM—A new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation. *J. Comp. Chem.* 15, 5 (1994), 488–506. 17
- [AW87] AMANATIDES J., WOO A.: A fast voxel traversal algorithm for ray tracing. In *Eurographics 1987 - Technical Papers* (1987), Eurographics Association, pp. 3–10. 11
- [Bad90] BADER R.: *Atoms in molecules*. Wiley & Sons, 1990. 16
- [BD12] BRANDT S., DAHMEN H. D.: *The picture book of quantum mechanics*. Springer, 2012. 2
- [BDF15] BARTOL T. M., DITTRICH M., FAEDER J. R.: MCell. In *Encyclopedia of Computational Neuroscience* (New York, 2015), Jaeger D., Jung R., (Eds.), Springer, pp. 1673–1676. 16
- [BDST04] BAJAJ C., DJEU P., SIDDAVANAHALLI V., THANE A.: Texmol: interactive visual exploration of large flexible multi-component molecular complexes. In *IEEE Visualization* (2004), pp. 243–250. 9, 10
- [Ber07] BERRY D.: Molecular animation of cell death mediated by the fas pathway. *Sci. Signal.* 2007, 380 (2007). 13
- [BGB\*08] BIDMON K., GROTTTEL S., BÖS F., PLEISS J., ERTL T.: Visual abstractions of solvent pathlines near protein cavities. *Comput. Graph. Forum* 27, 3 (2008), 935–942. 15
- [BHI\*09] BARTH I., HEGE H.-C., IKEDA H., KENFACK A., KOPPITZ M., MANZ J., MARQUARDT F., PARAMONOV G. K.: Concerted quantum effects of electronic and nuclear fluxes in molecules. *Chem. Phys. Lett.* 481, 1 (2009), 118–123. 16
- [BJG\*15] BYŠKA J., JURČÍK A., GRÖLLER M. E., VIOLA I., KOZLÍKOVÁ B.: MoleCollar and Tunnel Heat Map visualizations for conveying spatio-temporo-chemical properties across and along protein voids. *Comput. Graph. Forum* 34, 3 (2015), 1–10. 15
- [Blh77] BLINN J. F.: Models of light reflection for computer synthesized pictures. *Computer Graphics (Proc. SIGGRAPH 1977)* 11, 2 (1977), 192–198. 12

- [Bli82] BLINN J.: A generalization of algebraic surface drawing. *ACM Trans. Graph.* 1 (1982), 235–256. 8
- [Bor26] BORN M.: Zur Quantenmechanik der Stoßvorgänge. *Zeitschr. Phys.* 37, 12 (1926), 863–867. 2
- [BPG12] BRYDEN A., PHILLIPS G., GLEICHER M.: Automated illustration of molecular flexibility. *IEEE Trans. Vis. Comput. Graphics* 18, 1 (2012), 132–145. 14
- [BPW\*14] BERNIER C. R., PETROV A. S., WATERBURY C. C., JETT J., LI F., FREIL L. E., XIONG X., WANG L., MIGLIOZZI B. L. R., HERSHKOVITS E., XUE Y., HSIAO C., BOWMAN J. C., HARVEY S. C., GROVER M. A., WARTELL Z. J., WILLIAMS L. D.: RiboVision suite for visualization and analysis of ribosomes. *Faraday Discuss.* 169 (2014). 9
- [BR05] BAHAR I., RADER A.: Coarse-grained normal mode analysis in structural biology. *Curr. Opin. Struct. Biol.* 15, 5 (2005), 586–592. 4
- [BS03] BECKSTEIN O., SANSOM M. S.: Liquid-vapor oscillations of water in hydrophobic nanopores. *Proc. Natl. Acad. Sci. U.S.A.* 100, 12 (Jun 2003), 7063–7068. 15
- [Car91] CARSON M.: Ribbons 2.0. *J. Appl. Crystallogr.* 24, 5 (1991), 958–961. 9
- [CAZMS14] CHENG H.-C., ANGERMANN B. R., ZHANG F., MEIER-SCHELLERSHEIM M.: NetworkViewer: visualizing biochemical reaction networks with embedded rendering of molecular interaction rules. *BMC Syst. Biol.* 8, 70 (2014). 16
- [CB86] CARSON M., BUGG C. E.: Algorithm for Ribbon Models of Proteins. *J. Mol. Graph.* 4, 2 (1986), 121–122. 9
- [CBES11] CORTES J., BARBE S., ERARD M., SIMEON T.: Encoding molecular motions in voxel maps. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8, 2 (2011), 557–563. 14
- [CBS09] CHANDRAN D., BERGMANN F., SAURO H.: Tinker-Cell: modular CAD tool for synthetic biology. *J. Biol. Eng.* 3, 1 (2009), 19–36. 16
- [CCW06] CAN T., CHEN C.-I., WANG Y.-F.: Efficient molecular surface generation using level-set methods. *J. Mol. Graph. Modell.* 31, 4 (2006), 442–454. 6
- [CDB\*15] CRAGNOLINI T., DOUTRELIGNE S., BAADEN M., DERREUMAUX P., PASQUALI S.: 44 Predicting and exploring complex nucleic acids architectures through a coarse-grained model. *J. Biomol. Struct. Dyn.* 33 Suppl 1 (May 2015), 30–31. 4, 15
- [CG07] CIPRIANO G., GLEICHER M.: Molecular surface abstraction. *IEEE Trans. Vis. Comput. Graphics* 13, 6 (2007), 1608–1615. 10
- [CGJK\*11] CONTRERAS-GARCÍA J., JOHNSON E. R., KEINAN S., CHAUDRET R., PIQUEMAL J.-P., BERATAN D. N., YANG W.: NCIPLOT: a program for plotting noncovalent interaction regions. *J. Chem. Theory Comput.* 7, 3 (2011), 625–632. 16
- [CHF06] COUCH G. S., HENDRIX D. K., FERRIN T. E.: Nucleic acid visualization with UCSF Chimera. *Nucleic Acids Res.* 34, 4 (2006), e29. 9
- [CKH\*04] CRIVELLI S., KREYLOS O., HAMANN B., MAX N., BETHEL W.: ProteinShop: a tool for interactive protein manipulation and steering. *J. Comput. Aided Mol. Des.* 18, 4 (Apr. 2004), 271–285. 15
- [CKK\*05] COHEN J., KIM K., KING P., SEIBERT M., SCHULTEN K.: Finding gas diffusion pathways in proteins: Application to O<sub>2</sub> and H<sub>2</sub> transport in Cpl [FeFe]-hydrogenase and the role of packing defects. *Structure* 13, 9 (2005), 1321–1329. 14
- [CKSG09] CROSS S., KUTTEL M. M., STONE J. E., GAIN J. E.: Visualisation of cyclic and multi-branched molecules with VMD. *J. Mol. Graph. Modell.* 28, 2 (2009), 131–139. 10
- [Cle08] CLEMENTI C.: Coarse-grained models of protein folding: toy models or predictive tools? *Curr. Opin. Struct. Biol.* 18, 1 (2008), 10–15. 3, 10
- [CLK\*11] CHAVENT M., LEVY B., KRONE M., BIDMON K., NOMINE J.-P., ERTL T., BAADEN M.: GPU-powered tools boost molecular visualization. *Brief. Bioinform.* 12, 6 (2011), 689–701. 18
- [CLM08] CHAVENT M., LEVY B., MAIGRET B.: MetaMol: High quality visualization of molecular skin surface. *J. Mol. Graph. Modell.* 27, 2 (2008), 1391–1398. 8
- [Con83] CONNOLLY M. L.: Analytical molecular surface calculation. *J. Appl. Cryst.* 16, 5 (1983), 548–558. 6
- [CRG\*14] CHAVENT M., REDDY T., GOOSE J., DAHL A. C. E., STONE J. E., JOBARD B., SANSOM M. S. P.: Methodologies for the analysis of instantaneous lipid diffusion in MD simulations of large membrane systems. *Faraday Discuss.* 169 (2014), 455–475. 15
- [CS09] CHENG H.-L., SHI X.: Quality mesh generation for molecular skin surfaces using restricted union of balls. *Comput. Geom.* 42, 3 (2009), 196–206. 8
- [CVT\*11] CHAVENT M., VANEL A., TEK A., LEVY B., ROBERT S., RAFFIN B., BAADEN M.: GPU-accelerated atom and dynamic bond visualization using hyperballs: A unified algorithm for balls, sticks, and hyperboloids. *J. Comput. Chem.* 32, 13 (2011), 2924–2935. 5, 17
- [Dal10] DALTON J.: *A New System of Chemical Philosophy, Part I and II*. Birkenstaff, London, 1808 and 1810. Digital facsimile: <https://archive.org/details/newssystemofchemi01daltuoft>. 2
- [DCMP10] DOULAMIS N., CHRONIS E., MIAOULIS G., PLEMENOS D.: Personalized view selection of 3d molecular proteins. In *Intelligent Computer Graphics 2010*, Plemenos D., Miaoulis G., (Eds.), vol. 321. Springer, Berlin-Heidelberg, 2010, pp. 211–227. 13
- [DCS12] DAHL A. C. E., CHAVENT M., SANSOM M. S. P.: Bendix: intuitive helix geometry analysis and abstraction. *Bioinformatics* 28, 16 (2012), 2193–2194. 9
- [DeL02] DELANO W.: PyMOL: An open-source molecular graphics tool. *CCP4 Newsletter On Protein Crystallography* 40 (2002). <http://pymol.sourceforge.net/>. 9, 17
- [dHCKMK13] DE HERAS CIECHOMSKI P., KLANN M., MANGE R., KOEPL H.: From biochemical reaction networks to 3D dynamics in the cell: The ZigCell3D modeling, simulation and visualisation framework. In *IEEE Symposium on Biological Data Visualization* (2013), pp. 41–48. 16
- [DKM09] DENNINGTON R., KEITH T., MILLAM J.: GaussView Version 5, 2009. Semichem Inc., Shawnee Mission, KS. 17
- [DLB08] DURRIEU M. P., LAVERY R., BAADEN M.: Interactions between neuronal fusion proteins explored by molecular dynamics. *Biophys. J.* 94, 9 (May 2008), 3436–3446. 15
- [DPT\*13] DREHER M., PIUZZI M., TURKI A., CHAVENT M., BAADEN M., FÉREY N., LIMET S., RAFFIN B., ROBERT S.: Interactive molecular dynamics: Scaling up to large systems. *Procedia Comput. Sci.* 18 (2013), 20–29. 15
- [DR13] DECHERCHI S., ROCCHIA W.: A general and robust ray-casting-based algorithm for triangulating surfaces at the nanoscale. *PLoS ONE* 8, 4 (2013), e59744. 7, 8
- [DRSR15] DABDOUB S. M., RUMPF R. W., SHINDHELM A. D., RAY W. C.: MoFlow: visualizing conformational changes in molecules as molecular flow improves understanding. In *BMC Proc.* (2015), vol. 9, suppl 6, BioMed Central Ltd, p. S5. 14

- [Ede99] EDELSBRUNNER H.: Deformable smooth surface design. *Discrete Comput. Geom.* 21, 1 (1999), 87–115. 7
- [EHS13] EICHELBAUM S., HLAWITSCHKA M., SCHEUERMANN G.: LineAO - improved three-dimensional line rendering. *IEEE Trans. Vis. Comput. Graphics* 19, 3 (2013), 433–445. 13
- [EKK\*14] ERTL T., KRONE M., KESSELHEIM S., SCHARNOWSKI K., REINA G., HOLM C.: Visual analysis for space-time aggregation of biomolecular simulations. *Faraday Discuss.* 169 (2014), 167–178. 15
- [EM94] EDELSBRUNNER H., MÜCKE E. P.: Three-dimensional alpha shapes. *ACM Trans. Graph.* 13, 1 (1994), 43–72. 7
- [ESH13] EICHELBAUM S., SCHEUERMANN G., HLAWITSCHKA M.: PointAO - improved ambient occlusion for point-based visualization. In *EuroVis 2013 - Short Paper* (2013), pp. 13–17. 13
- [FKE13] FALK M., KRONE M., ERTL T.: Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. *Comput. Graph. Forum* 32, 8 (2013), 195–206. 4, 5, 11, 13
- [FKRE09] FALK M., KLANN M., REUSS M., ERTL T.: Visualization of signal transduction processes in the crowded environment of the cell. In *IEEE Pacific Visualization Symposium (PacificVis)* (2009), pp. 169–176. 16
- [FKRE10] FALK M., KLANN M., REUSS M., ERTL T.: 3d visualization of concentrations from stochastic agent-based signal transduction simulations. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (2010), pp. 1301–1304. 4, 14
- [FMKT03] FUNAHASHI A., MOROHASHI M., KITANO H., TANIMURA N.: CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO* 1, 5 (2003), 159–162. 16
- [FPE\*89] FUCHS H., POULTON J., EYLES J., GREER T., GOLDFEATHER J., ELLSWORTH D., MOLNAR S., TURK G., TEBBS B., ISRAEL L.: Pixel-planes 5: A heterogeneous multi-processor graphics system using processor-enhanced memories. In *SIGGRAPH '89* (1989), ACM, pp. 79–88. 5
- [Fra02] FRANCOEUR E.: Cyrus Levinthal, the Kluge and the origins of interactive molecular graphics. *Endeavour* 26, 4 (2002), 127–131. 1, 15
- [FSTR13] FIORAVANTE M., SHOOK A., THORPE I., RHEINGANS P.: Visualizing Motional Correlations in Molecular Dynamics using Geometric Deformations. *Comput. Graph. Forum* 32, 3pt3 (2013), 311–320. 14
- [FTS\*09] FRISCH M. J., TRUCKS G. W., SCHLEGEL H. B., SCUSERIA G. E., ROBB M. A., CHEESEMAN J. R., SCALMANI G., BARONE V., MENNUCCI B., PETERSSON G. A., ET AL.: Gaussian 09 Revision D.01, 2009. Gaussian Inc., Wallingford CT 2009. 17
- [GB78] GREER J., BUSH B. L.: Macromolecular shape and surface maps by solvent exclusion. *Proc. Natl. Acad. Sci. USA* 75 (1978), 303–307. 6
- [GBCG\*14] GÜNTHER D., BOTO R., CONTRERAS-GARCIA J., PIQUEMAL J.-P., TIERNY J.: Characterizing molecular interactions in chemical systems. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (2014), 2476–2485. 16
- [GF07] GODDARD T. D., FERRIN T. E.: Visualization software for molecular assemblies. *Curr. Opin. Struct. Biol.* 17, 5 (2007), 587–95. 9
- [GHZ\*15] GOMEZ T., HERMANN G., ZARATE X., PÉREZ-TORRES J. F., TREMBLAY J. C.: Imaging the ultrafast photo-electron transfer process in alizarin-tio2. *Molecules* 20, 8 (2015), 13830–13853. 16
- [Gil07] GILLESPIE D. T.: Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.* 58 (2007), 35–55. 4
- [GIL\*10] GRUENERT G., IBRAHIM B., LENSER T., LOHEL M., HINZE T., DITTRICH P.: Rule-based spatial modeling with diffusing, geometrically constrained molecules. *BMC Bioinformatics* 11, 307 (2010). 16
- [GKM\*15] GROTTTEL S., KRONE M., MÜLLER C., REINA G., ERTL T.: MegaMol - a prototyping framework for particle-based visualization. *IEEE Trans. Vis. Comput. Graphics* 21, 2 (2015), 201–214. 12, 13, 17
- [GKSE12] GROTTTEL S., KRONE M., SCHARNOWSKI K., ERTL T.: Object-space ambient occlusion for molecular dynamics. In *IEEE Pacific Visualization Symposium (PacificVis)* (2012), pp. 209–216. 12, 17
- [GMG08] GABRIEL A., MEYER T., GERMANO G.: Molecular graphics of convex body fluids. *J. Chem. Theory Comput.* 4 (2008), 468–476. 9
- [GO92] GOODSSELL D. S., OLSON A. J.: Molecular illustration in black and white. *J. Mol. Graph.* 10 (1992), 235–240. 12
- [Goo] GOODSSELL D. S.: Molecule of the month. [www.rcsb.org/pdb/motm.do](http://www.rcsb.org/pdb/motm.do), (online, Oct 2015). 12
- [Goo99] GOODSSELL D. S.: Atomistic vs. continuous representations in molecular biology. In *Visual Representations and Interpretations*, Paton R., Neilson I., (Eds.). Springer, London, 1999, pp. 146–155. 5
- [Goo09] GOODSSELL D. S.: *The Machinery of Life*, second ed. Springer, 2009. 3, 12
- [GP95] GRANT J. A., PICKUP B. T.: A Gaussian description of molecular shape. *J. Phys. Chem.* 99, 11 (1995), 3503–3510. 8
- [GRDE10] GROTTTEL S., REINA G., DACHSBACHER C., ERTL T.: Coherent culling and shading for large molecular dynamics visualization. *Comput. Graph. Forum* 29 (2010), 953–962. 11, 12, 17
- [GRE09] GROTTTEL S., REINA G., ERTL T.: Optimized data transfer for time-dependent, GPU-based glyphs. In *IEEE Pacific Visualization Symposium (PacificVis)* (2009), pp. 65–72. 6
- [GS87] GIBSON K., SCHERAGA H.: Exact calculation of the volume and surface area of fused hard-sphere molecules with unequal atomic radii. *Mol. Phys.* 62, 5 (1987), 1247–1265. 6
- [Gum03] GUMHOLD S.: Splatting illuminated ellipsoids with depth correction. In *Vision, Modeling, and Visualization* (2003), pp. 245–252. 5
- [Har02] HARTSHORN M. J.: AstexViewer: a visualisation aid for structure-based drug design. *J. Comput. Aided Mol. Des.* 16, 12 (Dec 2002), 871–881. 18
- [HCL\*12] HANWELL M. D., CURTIS D. E., LONIE D. C., VANDERMEERSCH T., ZUREK E., HUTCHISON G. R.: Avogadro: An advanced semantic chemical editor, visualization, and analysis platform. *J. Cheminform.* 4, 1 (2012), 17. 17
- [HDS96] HUMPHREY W., DALKE A., SCHULTEN K.: VMD: Visual molecular dynamics. *J. Mol. Graph.* 14 (1996), 33–38. 4, 9, 10, 17
- [Heh03] HEHRE W. J.: *A Guide to Molecular Mechanics and Quantum Chemical Calculations*. Wavefunction, 2003. 16
- [Hei26] HEISENBERG W.: Quantenmechanik. *Naturwiss.* 14, 45 (1926), 989–994. 2
- [HG08] HARANCZYK M., GUTOWSKI M.: Visualization of molecular orbitals and the related electron densities. *J. Chem. Theory Comput.* 4, 5 (2008), 689–693. 16

- [HGVV15] HERMOSILLA P., GUALLAR V., VÁZQUEZ P.-P., VINACUA A.: Adaptive on-the-fly molecular ribbons generation. In *EuroVis Poster Proceedings* (2015). 9
- [HK14] HASS J., KOEHL P.: How round is a protein? Exploring protein structures for globularity using conformal mapping. *Mathematics of Biomolecules 1* (2014), 26. 10
- [HKM\*11] HEGE H.-C., KOPPITZ M., MARQUARDT F., McDONALD C., MIELACK C.: Visual analysis of quantum physics data. In *Quantum Dynamic Imaging*. Springer, 2011, pp. 71–87. 16
- [HKOW14] HEINRICH J., KRONE M., O'DONOGHUE S. I., WEISKOPF D.: Visualising intrinsic disorder and conformational variation in protein ensembles. *Faraday Discuss. 169* (2014), 179–193. 14
- [HOF04] HALM A., OFFEN L., FELLNER D.: Visualization of complex molecular ribbon structures at interactive rates. In *Int. Conf. on Information Visualisation* (2004), pp. 737–744. 9
- [HPR\*13] HANSON R. M., PRILUSKY J., RENJIAN Z., NAKANE T., SUSSMAN J. L.: JSmol and the Next-Generation Web-Based Representation of 3D Molecular Structure as Applied to Proteopedia. *Isr. J. Chem. 53*, 3-4 (2013), 207–216. 18
- [Huy90] HUYGENS C.: *Traité de la lumière*. Pierre van der Aa, Leiden, 1690. Digital facsimile: <http://gallica.bnf.fr/ark:/12148/btv1b86069987>. 2
- [HVB\*14] HAAG M. P., VAUCHER A. C., BOSSON M., REDON S., REIHER M.: Interactive chemical reactivity exploration. *ChemPhysChem 15*, 15 (Oct 2014), 3301–3319. 15
- [Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, NY, USA, 2009. 14
- [Iwa08] IWASA J. H.: Animating the model figure. *Cell 20*, 12 (2008), 699–704. 13
- [JAAA\*15] JOHNSON G. T., AUTIN L., AL-ALUSI M., GOODSELL D. S., SANNER M. F., OLSON A. J.: cellPACK: a virtual mesoscope to model and visualize structural systems biology. *Nat. Methods 12*, 1 (2015), 85–91. 4, 17
- [JH14] JOHNSON G. T., HERTIG S.: A guide to the visual analysis and communication of biomolecular structural data. *Nat. Rev. Mol. Cell Biol. 15*, 10 (Oct 2014), 690–698. 13
- [JKMS\*10] JOHNSON E. R., KEINAN S., MORI-SÁNCHEZ P., CONTRERAS-GARCÍA J., COHEN A. J., YANG W.: Revealing noncovalent interactions. *J. Am. Chem. Soc. 132*, 18 (2010), 6498–6506. 16
- [jmo09] Jmol: an open-source Java viewer for chemical structures in 3D, 2009. [www.jmol.org](http://www.jmol.org), online Oct 2015. 18
- [Kab78] KABSCH W.: A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. A 34*, 5 (1978), 827–828. 15
- [Kaj09] KAJALIN V.: Screen-space ambient occlusion. In *ShaderX<sup>7</sup>*, Engel W., (Ed.). Charles River Media, 2009, ch. 6.1, pp. 413–424. 12
- [KBE08] KRONE M., BIDMON K., ERTL T.: GPU-based visualisation of protein secondary structure. In *Theory and Practice of Computer Graphics* (2008), vol. 8, The Eurographics Association, EG UK, pp. 115–122. 9, 17
- [KBE09] KRONE M., BIDMON K., ERTL T.: Interactive visualization of molecular surface dynamics. *IEEE Trans. Vis. Comput. Graphics 15*, 6 (2009), 1391–1398. 7, 12, 17
- [KBW96] KORADI R., BILLETER M., WÜTHRICH K.: MOLMOL: A program for display and analysis of macromolecular structures. *J. Mol. Graph. 14*, 1 (1996), 51–55. 13
- [KCL\*13] KNOLL A., CHAN M. K., LAU K. C., LIU B., GREELEY J., CURTISS L., HERELD M., PAKKA M. E.: Uncertainty classification and visualization of molecular interfaces. *Int. J. Uncertain. Quant. 3*, 2 (2013), 157–169. 16
- [Kep11] KEPLERI I.: *Strena, Seu de Nive Sexangula*. Francofurti ad Moenum, 1611. Digital facsimile: <https://archive.org/details/ioanniskepleriss00kepl>. 2
- [KFSR15] KOTTRAVEL S., FALK M., SUNDÉN E., ROPINSKI T.: Coverage-based opacity estimation for interactive depth of field in molecular visualization. In *IEEE Pacific Visualization Symposium (PacificVis)* (2015). To appear. 13
- [KG00] KREBS W., GERSTEIN M.: The morph server: a standardized system for analyzing and visualizing macromolecular motions in a database framework. *Nucleic Acids Res. 28*, 8 (Apr 2000), 1665–1675. 13
- [KGE11] KRONE M., GROTTTEL S., ERTL T.: Parallel contour-buildup algorithm for the molecular surface. In *IEEE Symposium on Biological Data Visualization* (2011), pp. 17–22. 7, 17
- [KH23] KRAMERS H. A., HOLST H.: *The Atom and the Bohr Theory of its Structure*. Gyldendal, London, 1923. 2
- [KK12] KREISBECK C., KRAMER T.: Long-lived electronic coherence in dissipative exciton dynamics of light-harvesting complexes. *The Journal of Physical Chemistry Letters 3*, 19 (2012), 2828–2833. 16
- [KKL\*15] KOZLÍKOVÁ B., KRONE M., LINDOW N., FALK M., BAADE M., BAUM D., VIOLA I., PARULEK J., HEGE H.-C.: Visualization of Biomolecular Structures: State of the Art. In *Eurographics Conference on Visualization (EuroVis) - STARs* (2015), Borgo R., Ganovelli F., Viola I., (Eds.), Eurographics Association. 1
- [KKP\*13] KAUKER D., KRONE M., PANAGIOTIDIS A., REINA G., ERTL T.: Rendering molecular surfaces using order-independent transparency. In *Eurographics Symposium on Parallel Graphics and Visualization* (2013), vol. 13, pp. 33–40. 7
- [KMH01] KOSARA R., MIKSCH S., HAUSER H.: Semantic depth of field. In *IEEE Symposium on Information Visualization (INFOVIS)* (2001), pp. 97–104. 13
- [KMH\*03] KREYLOS O., MAX N., HAMANN B., CRIVELL S., BETHEL E.: Interactive protein manipulation. In *IEEE Visualization* (2003), pp. 581–588. 15
- [KPV\*14] KOLESÁR I., PARULEK J., VIOLA I., BRUCKNER S., STAVRUM A.-K., HAUSER H.: Interactively illustrating polymerization using three-level model fusion. *BMC Bioinformatics 15*, 345 (2014), 1–16. 16
- [KSES12] KRONE M., STONE J. E., ERTL T., SCHULTEN K.: Fast visualization of Gaussian density surfaces for molecular dynamics and particle system trajectories. In *EuroVis - Short Papers* (2012), vol. 1, pp. 67–71. 8, 10, 17
- [KSN08] KANAMORI Y., SZEGO Z., NISHITA T.: GPU-based fast ray casting for a large number of metaballs. *Comput. Graph. Forum 27*, 2 (2008), 351–360. 8
- [KSS\*14] KOZLIKOVA B., SEBESTOVA E., SUSTR V., BREZOVSKY J., STRNAD O., DANIEL L., BEDNAR D., PAVELKA A., MANAK M., BEZDEKA M., BENES P., KOTRY M., GORA A., DAMBORSKY J., SOCHOR J.: CAVER Analyst 1.0: Graphic tool for interactive visualization and analysis of tunnels and channels in protein structures. *Bioinformatics* (2014). 4, 17
- [KV07] KRUIHOF N., VEGTER G.: Meshing skin surfaces with certified topology. *Computational Geometry 36*, 3 (2007), 166–182. 8
- [KV14] KRIEGER E., VRIEND G.: YASARA view—molecular

- graphics for all devices—from smartphones to workstations. *Bioinformatics* 30, 20 (2014), 2981–2982. 17, 18
- [KV15] KRIEGER E., VRIEND G.: New ways to boost molecular dynamics simulations. *J. Comput. Chem.* 36, 13 (2015), 996–1007. 3
- [KWN\*14] KNOLL A., WALD I., NAVRÁTIL P. A., BOWEN A., REDA K., PAKA M. E., GAITHER K.: RBF volume ray casting on multicore and manycore CPUs. *Comput. Graph. Forum* 33, 3 (2014), 71–80. 18
- [LB02] LAUG P., BOROUCHAKI H.: Molecular surface modeling and meshing. *Eng. Comput.* 18, 3 (2002), 199–210. 7
- [LBBH12] LINDOW N., BAUM D., BONDAR A.-N., HEGE H.-C.: Dynamic channels in biomolecular systems: Path analysis and visualization. In *IEEE Symposium on Biological Data Visualization* (2012), pp. 99–106. 15
- [LBH12] LINDOW N., BAUM D., HEGE H.-C.: Interactive rendering of materials and biological structures on atomic and nanoscopic scale. *Comput. Graph. Forum* 31 (2012), 1325–1334. 4, 5, 11, 12
- [LBH14] LINDOW N., BAUM D., HEGE H.-C.: Ligand excluded surface: A new type of molecular surface. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (2014), 2486–2495. 8
- [LBPH10] LINDOW N., BAUM D., PROHASKA S., HEGE H.-C.: Accelerated visualization of dynamic molecular surfaces. *Comput. Graph. Forum* 29, 3 (2010), 943–952. 7, 8, 18
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *ACM SIGGRAPH '87* (1987), vol. 21, pp. 163–169. 6
- [LCD06] LUFT T., COLDITZ C., DEUSSEN O.: Image enhancement by unsharp masking the depth buffer. *ACM Trans. Graph.* 25, 3 (2006), 1206–1213. 13
- [Lev66] LEVINTHAL C.: Molecular model-building by computer. *Sci. Am.* 214, 6 (1966), 42–52. 1
- [LKEP14] LAWONN K., KRONE M., ERTL T., PREIM B.: Line integral convolution for real-time illustration of molecular surface shape and salient regions. *Comput. Graph. Forum* 33, 3 (2014), 181–190. 12
- [LLNW14] LI H., LEUNG K.-S., NAKANE T., WONG M.-H.: iview: an interactive WebGL visualizer for protein-ligand complex. *BMC Bioinformatics* 15, 1 (2014), 56–62. 18
- [LMPSV14] LE MUZIC M., PARULEK J., STAVRUM A., VIOLA I.: Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. *Comput. Graph. Forum* 33, 3 (2014), 141–150. 11, 16
- [LN\*09] LE NOVÉRE N., ET AL.: The systems biology graphical notation. *Nat. Biotechnology* 27, 8 (2009), 735–741. 16
- [LPK06] LEE J., PARK S., KIM J.-I.: View-dependent rendering of large-scale molecular models using level of detail. In *Int. Conf. on Hybrid Information Technology (ICHIT)* (2006), vol. 1, pp. 691–698. 10
- [LR71] LEE B., RICHARDS F. M.: The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Bio.* 55, 3 (1971), 379–380. 2, 6
- [LTDS\*13] LV Z., TEK A., DA SILVA F., EMPEREUR-MOT C., CHAVENT M., BAADEN M.: Game on, science - how video game technology may help biologists tackle visualization challenges. *PLoS ONE* 8, 3 (2013), e57990. 4, 14, 17
- [LV02] LEE C. H., VARSHNEY A.: Representing thermal vibrations and uncertainty in molecular surfaces. In *SPIE Conference on Visualization and Data Analysis* (2002), pp. 80–90. 14
- [LVRH07] LAMPE O. D., VIOLA I., REUTER N., HAUSER H.: Two-level approach to efficient visualization of protein dynamics. *IEEE Trans. Vis. Comput. Graphics* 13, 6 (2007), 1616–1623. 11, 12
- [MAPV15] MUZIC M. L., AUTIN L., PARULEK J., VIOLA I.: cellVIEW: a tool for illustrative and multi-scale rendering of large biomolecular datasets. In *EG Workshop on Visual Computing for Biology and Medicine* (Sept. 2015), pp. 11
- [McG08] MCGILL G.: Molecular movies... coming to a lecture near you. *Cell* 133, 7 (Jun 2008), 1127–1132. 13
- [MEP92] MOLNAR S., EYLES J., POULTON J.: Pixelflow: High-speed rendering using image composition. In *SIGGRAPH '92* (1992), ACM, pp. 231–240. 5
- [MHLK05] MOLL A., HILDEBRANDT A., LENHOF H.-P., KOHLBACHER O.: BALLView: A tool for research and education in molecular modeling. *Bioinformatics* (2005). 18
- [Mi94] MILLER G.: Efficient algorithms for local and global accessibility shading. In *SIGGRAPH '94* (1994), ACM, pp. 319–326. 12
- [MKB\*15] MWALONGO F., KRONE M., BECHER M., REINA G., ERTL T.: Remote Visualization of Dynamic Molecular Data Using WebGL. In *International Conference on 3D Web Technology* (2015), ACM, pp. 115–122. 18
- [MKK\*14] MWALONGO F., KRONE M., KARCH G., BECHER M., REINA G., ERTL T.: Visualization of Molecular Structures using State-of-the-Art Techniques in WebGL. In *International Conference on 3D Web Technology* (2014), vol. 19, ACM, pp. 133–141. 18
- [MM04] MORITZ E., MEYER J.: Interactive 3D protein structure visualization using virtual reality. In *IEEE Symposium on Bioinformatics and Bioengineering* (2004), pp. 503–507. 9
- [mma13] Molecular Maya Toolkit – mMaya. <http://www.molecularmovies.com/toolkit/>, 2013. 17
- [MSS\*08] MORARU I., SCHAFF J., SLEPCHENKO B., BLINOV M., MORGAN F., LAKSHMINARAYANA A., GAO F., LI Y., LOEW L.: Virtual cell modelling and simulation software environment. *IET Syst. Biol.* 2, 5 (2008), 352–362. 16
- [MWPV15] MUZIC M. L., WALDNER M., PARULEK J., VIOLA I.: Illustrative Timelapse: A Technique for Illustrative Visualization of Particle Simulations on the Mesoscale Level. In *IEEE Pacific Visualization Symposium* (2015), pp. 247–254. 4, 16
- [NCS85] NAMBA K., CASPAR D. L., STUBBS G. J.: Computer graphics representation of levels of organization in tobacco mosaic virus structure. *Science* 227, 4688 (1985), 773–776. 9
- [NHK\*95] NELSON M., HUMPHREY W., KUFIRIN R., GURSOY A., DALKE A., KALE L., SKEEL R., SCHULTEN K.: MDScope – a visual computing environment for structural biology. *Comput. Phys. Commun.* 91, 1–3 (1995), 111 – 133. 15
- [PB13] PARULEK J., BRAMBILLA A.: Fast blending scheme for molecular surface representation. *IEEE Trans. Vis. Comput. Graphics* 19, 12 (2013), 2653–2662. 8
- [PC51] PAULING L., COREY R. B.: The pleated sheet, a new layer configuration of polypeptide chains. *Proc. Natl. Acad. Sci. USA* 37, 5 (1951), 251–256. 3
- [PCB51] PAULING L., COREY R. B., BRANSON H. R.: The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. USA* 37, 4 (1951), 205–211. 3
- [PGH\*04] PETERSEN E. F., GODDARD T. D., HUANG C. C., COUCH G. S., GREENBLATT D. M., MENG E. C., FERRIN T. E.: UCSF Chimera - a visualization system for exploratory



- research and analysis. *J. Comput. Chem.* 25, 13 (2004), 1605–1612. [17](#)
- [Pho75] PHONG B. T.: Illumination for computer generated pictures. *Commun. ACM* 18, 6 (1975), 311–317. [12](#)
- [PJR\*14] PARULEK J., JÖNSSON D., ROPINSKI T., BRÜCKNER S., YNNERMAN A., VIOLA I.: Continuous levels-of-detail and visual abstraction for seamless molecular visualization. *Comput. Graph. Forum* 33, 6 (2014), 276–287. [4](#), [12](#)
- [PL00] PORTMANN S., LUTHI H. P.: MOLEKEL: An interactive molecular graphics tool. *Chimia* 54, 12 (2000), 766–770. [17](#)
- [PS05] PLIMPTON S. J., SLEPOY A.: Microbial cell modeling via reacting diffusive particles. *J. Phys.: Conference Series* 16 (2005), 305–309. [4](#)
- [PS09] POSTARNAKEVICH N., SINGH R.: Global-to-local representation and visualization of molecular surfaces using deformable models. In *ACM Symposium on Applied Computing* (2009), pp. 782–787. [10](#)
- [PS15] PATANE G., SPAGNUOLO M.: State-of-the-art and perspectives of geometric and implicit modeling for molecular surfaces. In *Computational Electrostatics for Biological Applications*, Rocchia W., Spagnuolo M., (Eds.). Springer, 2015, pp. 157–176. [7](#)
- [PTIB14] PÉREZ S., TUBIANA T., IMBERTY A., BAADEN M.: Three-dimensional representations of complex carbohydrates and polysaccharides-SweetUnityMol: A video game-based computer graphic software. *Glycobiology* (2014). [10](#), [17](#)
- [PV12] PARULEK J., VIOLA I.: Implicit representation of molecular surfaces. In *IEEE Pacific Visualization Symposium (PacificVis)* (2012), pp. 217–224. [7](#)
- [RBT14] ROZMANOV D., BAOUKINA S., TIELEMAN D. P.: Density based visualization for molecular simulation. *Faraday Discuss.* 169, 0 (2014), 225–243. [14](#)
- [RCDF08] RUSINKIEWICZ S., COLE F., DECARLO D., FINKELSTEIN A.: Line drawings from 3D models. In *ACM SIGGRAPH 2008 Classes* (2008), vol. 39, pp. 1–356. [12](#)
- [RCK09] RYU J., CHO Y., KIM D.-S.: Triangulation of molecular surfaces. *Comput. Aided Des.* 41, 6 (2009), 463–478. [7](#)
- [RE05] REINA G., ERTL T.: Hardware-accelerated glyphs for mono- and dipoles in molecular dynamics visualization. In *EG/IEEE VGTC Symposium on Visualization* (2005), pp. 177–182. [5](#)
- [RFD\*10] RIALLE S., FELICORI L., DIAS-LOPES C., PÉRÈS S., EL ATIA S., THIERRY A. R., AMAR P., MOLINA F.: BioNetCAD: design, simulation and experimental validation of synthetic biochemical networks. *Bioinformatics* 26, 18 (2010), 2298–2304. [16](#)
- [RH15] ROSE A., HILDEBRAND P.: NGL Viewer: a web application for molecular visualization. *Nucleic Acids Res.* 43, 5 (2015), W576–9. [18](#)
- [Ric77] RICHARDS F. M.: Areas, volumes, packing, and protein structure. *Ann. Rev. Biophys. Bio.* 6, 1 (1977), 151–176. [6](#)
- [Ric81] RICHARDSON J.: The anatomy and taxonomy of protein structure. *Adv. Protein Chem.* 34 (1981), 167–339. [9](#)
- [RJ99] RHEINGANS P., JOSHI S.: Visualization of molecules with positional uncertainty. In *Data Visualization '99* (1999), pp. 299–306. [13](#)
- [RK15] REGO N., KOES D.: 3Dmol.js: molecular visualization with WebGL. *Bioinformatics* 31, 8 (2015), 1322–1324. [18](#)
- [RRS63] RAMACHANDRAN G., RAMAKRISHNAN C., SASISEKHARAN V.: Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.* 7, 1 (1963), 95–99. [9](#)
- [RS07] RAHI S. J., SHARP K.: Mapping complicated surfaces onto a sphere. *Int. J. Comput. Geom. Ap.* 17, 04 (2007), 305–329. [10](#)
- [RSL13] ROBERTS E., STONE J. E., LUTHEY-SCHULTEN Z.: Lattice Microbes: high-performance stochastic simulation method for the reaction-diffusion master equation. *J. Comput. Chem.* 34, 3 (Jan 2013), 245–255. [4](#)
- [SB01] STILES J. R., BARTOL T. M.: *Monte Carlo Methods for Simulating Realistic Synaptic Microphysiology Using MCell*. CRC Press, 2001, ch. 4. [4](#)
- [SBB\*93] SCHMIDT M. W., BALDRIDGE K. K., BOATZ J. A., ELBERT S. T., GORDON M. S., JENSEN J. H., KOSEKI S., MATSUNAGA N., NGUYEN K. A., SU S., ET AL.: General atomic and molecular electronic structure system. *J. Comput. Chem.* 14, 11 (1993), 1347–1363. [17](#)
- [Sch26] SCHRÖDINGER E.: Quantisierung als Eigenwertproblem - vierte Mitteilung. *Ann. Phys.* 81, 18 (1926), 109–139. [2](#)
- [Sch10] SCHLICK T.: *Molecular Modeling and Simulation: An Interdisciplinary Guide*, vol. 21. Springer, 2010. [3](#)
- [SEBH02] SCHMIDT-EHRENBERG J., BAUM D., HEGE H.-C.: Visualizing dynamic molecular conformations. In *IEEE Visualization* (2002), pp. 235–242. [13](#)
- [SF02] SMIT B., FRENKEL D.: *Understanding molecular simulation: from algorithms to applications*. Elsevier Science & Technology, 2002. [3](#)
- [SGB\*14] SHAW D. E., GROSSMAN J. P., BANK J. A., BATSON B., ET AL.: Anton 2: Raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In *Int. Conf. for High Performance Computing, Networking, Storage and Analysis* (2014), SC'14, pp. 41–53. [3](#)
- [SGG15] STAIB J., GROTTTEL S., GUMHOLD S.: Visualization of particle-based data with transparency and ambient occlusion. *Comput. Graph. Forum* 34, 3 (2015), 151–160. [13](#)
- [SGSG01] STONE J. E., GULLINGSRUD J., SCHULTEN K., GRAYSON P.: A system for interactive molecular dynamics simulation. In *ACM Symposium on Interactive 3D Graphics* (2001), pp. 191–194. [15](#)
- [She99] SHERSTYUK A.: Kernel functions in convolution surfaces: a comparative analysis. *The Visual Computer* 15, 4 (1999), 171–182. [8](#)
- [SHLK11] STONE J. E., HARDY D. J., L. V. K., K. S.: GPU-accelerated computation and interactive display of molecular orbitals. In *GPU Computing Gems Emerald Edition*, Hwu W.-m. W., (Ed.). Morgan Kaufmann, 2011, ch. 1, pp. 5–18. [17](#)
- [SI12] SZECSI L., ILLES D.: Real-time metaball ray casting with fragment lists. In *EG 2012 Short Papers* (2012), pp. 93–96. [8](#)
- [SKS\*13] SCHARNOWSKI K., KRONE M., SADLO F., BECK P., ROTH J., TREBIN H.-R., ERTL T.: 2012 IEEE Visualization Contest Winner: Visualizing polarization domains in barium titanate. *IEEE Comput. Graph. Appl.* 33, 5 (2013), 9–17. [14](#)
- [SKVS10] STONE J. E., KOHLMAYER A., VANDIVORT K. L., SCHULTEN K.: Immersive molecular visualization and interactive modeling with commodity hardware. In *Advances in Visual Computing*, vol. 6454 of *Lecture Notes in Computer Science*. Springer, 2010, pp. 382–393. [15](#)
- [Smi60] SMITH D. K.: *Bibliography on molecular and crystal structure models*. Tech. rep., US National Bureau of Standards, 1960. [2](#)
- [SMO\*03] SHANNON P., MARKIEL A., OZIER O., BALIGA N. S., WANG J. T., RAMAGE D., AMIN N., SCHWIKOWSKI B., IDEKER T.: Cytoscape: a software environment for integrated

- models of biomolecular interaction networks. *Genome Res.* 13, 11 (Nov 2003), 2498–2504. [16](#)
- [SMW95] SAYLE R. A., MILNER-WHITE E.: RASMOL: biomolecular graphics for all. *Trends Biochem. Sci.* 20, 9 (1995), 374–376. [6](#)
- [SN00] SCHAFTENAAR G., NOORDIK J.: Molden: a pre- and post-processing program for molecular and electronic structures. *J. Comput. Aided Mol. Des.* 14, 2 (2000), 123–134. [17](#)
- [SO97] SANNER M. F., OLSON A. J.: Real time surface reconstruction for moving molecular fragments. In *Pacific Symposium on Biocomputing* (1997), pp. 385–396. [7](#)
- [SOS96] SANNER M. F., OLSON A. J., SPEHNER J.-C.: Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers* 38, 3 (1996), 305–320. [6](#), [7](#)
- [SRRB94] SURLS M. C., RICHARDSON J. S., RICHARDSON D. C., BROOKS F. P.: Sculpting proteins interactively: continual energy minimization embedded in a graphical modeling system. *Protein Sci* 3, 2 (1994), 198–210. [15](#)
- [SSH\*09] STONE J. E., SAAM J., HARDY D. J., VANDIVORT K. L., HWU W.-M. W., SCHULTEN K.: High performance computation and interactive display of molecular orbitals on GPUs and multi-core CPUs. In *Workshop on General Purpose Processing on Graphics Processing Units* (2009), pp. 9–18. [16](#)
- [ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-D shapes. *Computer Graphics (Proc. SIGGRAPH 1990)* 24, 4 (1990), 197–206. [13](#)
- [Str95] STRYER L.: *Biochemistry*. W.H. Freeman and Company, New York, 1995. [3](#)
- [SVGR15] SKANBERG R., VAZQUEZ P.-P., GUALLAR V., ROPINSKI T.: Real-Time Molecular Visualization Supporting Diffuse Illumination and Ambient Occlusion. *IEEE Trans. Vis. Comput. Graphics* (2015). (to appear). [19](#)
- [SVS13] STONE J. E., VANDIVORT K. L., SCHULTEN K.: GPU-accelerated molecular visualization on petascale supercomputing platforms. In *Proc. 8th Int. Workshop on Ultrascale Visualization* (2013), UltraVis '13, ACM, pp. 6:1–6:8. [18](#)
- [SWBG06] SIGG C., WEYRICH T., BOTSCH M., GROSS M.: GPU-based ray-casting of quadratic surfaces. In *EG Symposium on Point-Based Graphics* (2006), pp. 59–65. [5](#), [11](#)
- [SWH05] STALLING D., WESTERHOFF M., HEGE H.-C.: Amira: a highly interactive system for visual data analysis. In *The Visualization Handbook*. Elsevier, 2005, pp. 749–767. [17](#)
- [SXS\*11] SMITH A., XU W., SUN Y., FAEDER J., MARAI G.: RuleBender: Integrated visualization for biochemical rule-based modeling. In *IEEE Symposium on Biological Data Visualization* (2011), pp. 1–8. [16](#)
- [TA96] TOTROV M., ABAGYAN R.: The contour-buildup algorithm to calculate the analytical molecular surface. *J. Struct. Biol.* 116, 1 (1996), 138–143. [7](#)
- [TCB\*12] TEK A., CHAVENT M., BAADEN M., DELALANDE O., BOURDOT P., FERAY N.: Advances in human-protein interaction—interactive and immersive molecular simulations. In *Protein-Protein Interactions—Computational and Experimental Tools* (2012), Cai W., Hong H., (Eds.), InTech. Chapter 2. [16](#)
- [TCM06] TARINI M., CIGNONI P., MONTANI C.: Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Trans. Vis. Comput. Graphics* 12, 5 (2006), 1237–1244. [12](#), [13](#), [17](#)
- [Tha05] THALLER B.: *Advanced visual quantum mechanics*. Springer, 2005. [2](#)
- [Toz05] TOZZINI V.: Coarse-grained models for proteins. *Curr. Opin. Struct. Biol.* 15, 2 (2005), 144–150. [10](#)
- [TWK\*11] THOMAS B., WALTER J., KRONE M., HASSE H., ERTL T.: Interactive exploration of polymer-solvent interactions. In *Vision, Modeling and Visualization* (2011), vol. 16, pp. 301–308. [14](#)
- [UD10] UVERSKY V. N., DUNKER A. K.: Understanding protein non-folding. *Biochimica et Biophysica Acta - Proteins and Proteomics* 1804, 6 (2010), 1231–1264. [14](#)
- [VBW94] VARSHNEY A., BROOKS F. P., WRIGHT W. V.: Linearly scalable computation of smooth molecular surfaces. *IEEE Comput. Graph. Appl.* 14, 5 (1994), 19–25. [7](#), [8](#)
- [vdW73] VAN DER WAALS J. D.: *Over de continuïteit van den gas-en vloeistoftoestand*. PhD thesis, Univ. Leiden, 1873. [2](#)
- [vdZLB11] VAN DER ZWAN M., LUEKS W., BEKKER H., ISENBERG T.: Illustrative molecular visualization with continuous abstraction. *Comput. Graph. Forum* 30, 3 (2011), 683–690. [4](#), [10](#), [12](#)
- [VFG98] VELHO L., FIGUEIREDO L. H. D., GOMES J. A.: *Implicit Objects in Computer Graphics*. Springer, New York, 1998. [8](#)
- [VFSL02] VÁZQUEZ P.-P., FEIXAS M., SBERT M., LLOBET A.: Viewpoint entropy: a new tool for obtaining good views of molecules. In *Symposium on Data Visualisation (VISSYM '02)* (2002), vol. 22, pp. 183–188. [13](#)
- [vHGM\*00] VAN HEEL M., GOWEN B., MATADEEN R., ORLOVA E. V., FINN R., PAPE T., COHEN D., STARK H., SCHMIDT R., SCHATZ M., ET AL.: Single-particle electron cryo-microscopy: towards atomic resolution. *Q. Rev. Biophys.* 33, 04 (2000), 307–369. [3](#)
- [VPL11] VEHLW C., PREIM B., LAPPE M.: Visualization of anisotropic contact potentials within protein structures. In *2011 IEEE Symposium on Biological Data Visualization (BioVis)* (2011), pp. 31–38. [9](#)
- [WB11] WAHLE M., BIRMANNS S.: GPU-accelerated visualization of protein dynamics in ribbon mode. In *SPIE Conference on Visualization and Data Analysis* (2011), pp. 786–805. [3](#), [9](#)
- [WC53] WATSON J., CRICK F.: A structure for deoxyribose nucleic acid. *Nature* 421, 6921 (1953), 397–3988. [3](#)
- [Web09] WEBER J. R.: ProteinShader: illustrative rendering of macromolecules. *BMC Struct. Biol.* 9, 19 (2009). [12](#), [17](#)
- [Woo97] WOOLFSON M. M.: *An introduction to X-ray crystallography*. Cambridge University Press, 1997. [3](#)
- [Wüt86] WÜTHRICH K.: *NMR of proteins and nucleic acids*. John Wiley & Sons, New York, 1986. [3](#)
- [Yu09] YU Z.: A list-based method for fast generation of molecular surfaces. In *Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (2009), vol. 31, pp. 5909–5912. [6](#)
- [ZIK98] ZHUKOV S., IONES A., KRONIN G.: An ambient light illumination model. In *Eurographics Workshop on Rendering* (1998), pp. 45–56. [12](#)
- [ZXB07] ZHAO W., XU G., BAJAJ C.: An algebraic spline model of molecular surfaces. In *Symposium on Solid and Physical Modeling (SPM)* (2007), pp. 297–302. [7](#)

## Author Biographies

**Marc Baaden**, research director at CNRS in Paris, has 14 years of post-PhD practice as a computational chemist. His research focuses on interactive molecular modelling approaches for biological systems. Molecular visualization, including virtual reality-based and game engine-based applications, is an essential part of this work.

**Daniel Baum** works as a senior researcher at Zuse Institute Berlin. He received his PhD degree from Freie Universität Berlin for his work on molecular visualization and similarity. His current interests include the visualization and analysis of image data as well as molecular structures.

**Martin Falk** is a postdoctoral researcher at Linköping University. He received his PhD degree from the University of Stuttgart in 2013. His research interests are molecular visualizations in the context of systems biology, spatio-temporal data, glyph-based rendering, and GPU-based simulations.

**Hans-Christian Hege** is head of the Visual Data Analysis department at Zuse Institute Berlin. He started research in computational physics at Freie Universität Berlin, then joined the High-Performance Computing division at ZIB and later built up the Scientific Visualization department. His research includes methods of visual computing and data analysis as well as applications in natural and life sciences.

**Barbora Kozlikova** is an assistant professor at the Masaryk University in Brno, where she also received her PhD degree in 2011. Her main research interests are geometric analysis and visualization of molecular structures. She is a member of the International Society for Computational Biology.

**Michael Krone** is a PhD candidate at the Visualization Research Center of the University of Stuttgart (VISUS), Germany. His main research interest are molecular visualization with a focus on molecular graphics for structural biology, particle rendering, and GPU-accelerated computing.

**Norbert Lindow** is a PhD student at Zuse Institute Berlin (ZIB). His research interests are computer graphics and computational geometry for interactive biomolecular visualization and structure analysis.

**Julius Parulek** is an adjunct associate professor with University of Bergen. He received his PhD degree in 2008 at Comenius University, Bratislava. His research mainly focuses on the visualization in biology and 3D geometrical modeling.

**Ivan Viola** is an assistant professor at the TU Wien, Austria and an adjunct professor at University of Bergen, Norway. He received M.Sc. in 2002 and Ph.D. in 2005 from TU Wien, Austria. His research is focusing on effective visualization methods that are well understandable by humans, in the context of several application domains, including structural biology.